



# ESCOLA NAVAL

*talant de bi-faire*



## **Classificação de navios Baseados em Assinaturas Acústicas**

*Elaboração de uma aplicação em Python de Classificação de  
gravações acústicas*

**Cláudio Tiago Miguel António**

**Dissertação para obtenção do grau de Mestre em Ciências**

**Militares Navais, na especialidade de Marinha**



**Alfeite**

**2019**





# ESCOLA NAVAL

ta sante bi faire



**Cláudio Tiago Miguel António**

## **Classificação de navios Baseados em Assinaturas Acústicas**

**Dissertação para obtenção do grau de Mestre em Ciências Militares Navais,  
na especialidade de Marinha**

Orientação de: Professor Doutor Ricardo Moura

Co-orientação: Professor Doutor Victor Lobo

**O Aluno Mestrando**

**O Orientador**

---

Cláudio Tiago Miguel António

---

Professor Doutor Ricardo Moura

**Alfeite**

**2019**



# Epígrafe

“Sempre parece impossível até que seja feito”

-Nelson Mandela (1918 – 2013)



# Dedicatória

Dedico este trabalho primeiramente a toda minha Família (Família Miguel) que sempre esteve ao meu lado em todos momentos durante o meu percurso académico e em especial à memória de meu Pai, Ceita Moniz, pois sei que estará a olhar por nós e certamente muito orgulhoso e à minha filha Aylla António.





# Agradecimentos

Primeiramente, gostaria de agradecer a Deus pela dádiva da vida. Agradeço também à minha mãe, Maria Miguel, pelo seu enorme esforço e dedicação, no passado e no presente, lutando por mim, aos meus irmãos queridos, não esquecendo a minha outra mãe, Ana Miguel, e aos meus outros queridos irmãos, Aylton, Patrícia e Débora, pela recepção e carinho.

Agradeço a todos os meus tios pelo suporte, conselhos e principalmente por depositarem a sua confiança em mim.

Agradeço imenso à minha querida noiva, Flávia Ant3nio, pelo apoio constante, e principalmente ao Professor Doutor Ricardo Moura pela orienta33o, disponibilidade e todas as chamadas de aten33o, bem como ao Professor Doutor Victor Lobo que me guiou para a realiza33o deste trabalho.

A todos meus amigos, que sempre estiveram ao meu lado desde o in3cio, um muito obrigado.



# Resumo

A presente dissertação tem como objetivo desenvolver uma aplicação na linguagem Python de um modelo de classificação de assinaturas acústicas (Navios), usando gravações fornecidas pelo Professor Catedrático Victor Lobo no âmbito da sua tese de Doutoramento.

A partir das gravações originais foram extraídas várias características áudio usando transformadas de Fourier de tempo curto, coeficientes cepstrais da frequência Mel, componentes do respetivo cromagrama, espectrograma em escala de Mel, contraste espectral e Tonnetz. À base de dados compilada com todas as características extraídas dos ficheiros áudio construiu-se um Mapa Auto-Organizado de Kohonen (SOM) usando um algoritmo *batch*. Usando o SOM como classificador avaliou-se a precisão de este classificar corretamente um grupo de áudios teste de navios e foi feita uma comparação com a precisão que se obteria usando uma rede neuronal, o perceptron de multicamadas.

De seguida, foi criado um sistema de decisão que auxiliasse a identificação do navio disponibilizando não só a classificação mais provável, como também as outras possibilidades de classificação.

# Abstract

The main objective of the present work is to develop a Python application of acoustic signatures classification (ships), using audio recordings given by Professor Doctor Victor Lobo., within the framework of his PhD thesis.

From original recordings several audio features were extracted using short time Fourier transform, Mel-frequency cepstral coefficients, features from respective chromagram, Mel-scaled spectrogram, spectral contrast and its Tonnetz. A batch Kohonen Self-Organizing Map (SOM) was then applied to the complete matrix of compiled data. Using the SOM as a classifier, it is evaluated the precision of this technique classifying correctly a group test of ships and it is compared with the precision obtained by using a neuronal network multi-layer perceptron.

Moreover, decision system that supports the ship identification not only making available the most likely classification, but also the other nearest possibilities.

# Índice

Epígrafe .....	I
Dedicatória .....	III
Agradecimentos .....	V
Resumo .....	VII
Abstract .....	VIII
Índice .....	IX
Índice de Tabelas .....	XI
Índice de Figuras .....	XII
1. Introdução .....	1
1.1 Motivação .....	2
1.2 Estrutura do Documento .....	3
2 Acústica Submarina .....	5
2.1 Influência do meio .....	6
2.1.1 Ruídos .....	7
2.2 Caraterísticas das Ondas Acústicas .....	8
2.2.1 Comprimento de onda ( $\lambda$ ), velocidade e frequência .....	9
2.2.2 Intensidade Acústica .....	10
2.3 Aplicações Militares .....	11
2.4 Aplicações Cíveis .....	12
3 Processamento de Sinal .....	15
3.1 Transformadas de Fourier .....	18
3.2 Transformada de Fourier em Tempo curto .....	20
3.3 Coeficientes Mel-Cepstrais .....	21
3.4 Componentes de Cromagrama .....	22
4 <i>Data Mining</i> .....	25
4.1 Pré-processamento dos dados .....	27
4.1.1 Normalização Min-Max .....	28
4.1.2 Normalização Standard .....	29
4.2 Técnicas de Agrupamento .....	30
4.2.1 K-médias .....	32
4.2.2 Mapas auto-organizados (SOM) .....	34
4.3 Técnicas de Previsão .....	39
4.3.1 Classificação .....	39
4.3.2 K-vizinhos .....	41
4.3.3 Redes Neurais Artificiais .....	43
5 Aplicação .....	49

6 Conclusões.....	55
Bibliografia.....	59
Anexos.....	65
Anexo A – Resultados de múltiplos SOM's .....	65
Anexo B – Tabela de ‘presença’ do SOM selecionado .....	67
Anexo C – Mapas de classificação dos SOM's .....	69
Anexo D – Resultado de múltiplas RNA's .....	75
Anexo E – Programas .....	79

# Índice de Tabelas

Tabela 1: Resultados de 10 mapas treinados, com dimensão $7 \times 10$ , grelha hexagonal, 20 iterações de treino bruto e 50 iterações de treino fino.....	51
Tabela 2: Comparação entre a Classificação Prevista, a Classificação Real e a 2. <sup>a</sup> Classificação mais provável dos dados mal classificados. ....	54

# Índice de Figuras

Figura 1: Pirâmide de Dados, Informação e Conhecimento .....	2
Figura 2: Representação de onda sonora ("PEEAS 43 (A)") .....	9
Figura 3: Diagrama Simplificado do DSP (Tan & Jeang, 2013).....	15
Figura 4: Sinal Analógico e os respectivas amostras digitais e respetivo Intervalo de Amostragem (Tan & Jeang, 2013, p. 16) .....	16
Figura 5: Sinal bem amostrado vs Sinal Mal Amostrado. (Tan & Jeang, 2013, p. 17)..	17
Figura 6: Representação de um sinal, decomposto em três sinais com diferentes frequências e a mesma amplitude, vistos no domínio do tempo e da frequência (Brandt, 2011, p. 170).....	18
Figura 7: Audio e Espectrograma de um segmento de um registo sonar de um navio. .	20
Figura 8: Passo para a extração do vector MFCC's (Majeed, Husain, & Samad, 2015) .....	21
Figura 9: Cromagrama de um Sinal de um registo de um navio. (Python™) .....	23
Figura 10: Etapas do Processo de KDD. (Camargo, 2016) .....	26
Figura 11: Passos para o processo de KDD (Simplificado). (Tan, Steinbach, & Kumar, 2005).....	27
Figura 12: Função de Densidade da Normal com $N(0,1)$ e $N(4,1)$ . (Duarte & Pereira, 1990).....	29
Figura 13: Dados simulados no plano agrupados em 3 classes pelo algoritmo k-médias. (Hastie, Tibshiran, & Friedman, 2008, p. 521) .....	31
Figura 14: Estrutura Básica de um Mapa Auto-organizável (SOM). (Lobo V. J., 2002) .....	35
Figura 15: Exemplo de representação de Mapas SOM's (Retangular e Hexagonal). (Gonçalves, L De, Netto, & Zullo Junior, 2019).....	36
Figura 16: Exemplo de dados de treino, validação e teste.....	40
Figura 17: Exemplo de uma Rede Neuronal Artificial organizada por camadas. ....	43
Figura 18: Representação Simplificada do Neurônio Biológico. (Haykin, 2001).....	44
Figura 19: Modelo não linear de um neurónio. (Haykin, 2001, p. 36).....	45
Figura 20: Transformação afim produzida pela presença de um bias. (Haykin, 2001, p. 37).....	46



Figura 21: Mapa de componentes contendo a classificação predominante como anotação. ....	52
Figura 22: Precisão de treino e Precisão validação (cima) Erro de Treino e Erro de Validação (baixo). ....	53



# 1. Introdução

Dados, informação e conhecimento são conceitos que lidamos diariamente nas nossas vidas, seja em casa, escola ou mesmo no trabalho, sendo importante conhecer a diferença que existe entre eles. Existem diversas definições para os conceitos acima citados, mas é importante salientar que, quer os dados, quer as informações, constituem ali-cerces para a construção do conhecimento pois os dados são definidos como sendo códigos que constituem a matéria prima da informação, ou seja, é a informação não tratada, números e outros itens elementares que não têm associado um significado específico, já a informação que é apresentada como o resultado da interpretação ou do processamento de dados (Alavi, 2001) é a informação que fornece sentido aos dados de forma a obter descrições de acontecimentos, objetos ou situações. O conhecimento é definido como sendo a informação personalizada, ou seja, informação que as pessoas possam usar para dar significado aos acontecimentos, objetos e situações (Santos & Ramos, 2017).

Podemos verificar assim que, para cada tipo de problema existem diversos dados e na presente Tese pretende-se ajudar a solucionar o problema que os submarinos enfrentam para identificar navios na sua proximidade a partir do som subaquático por eles produzido construindo um modelo de classificação que irá carregar os dados (sons), extrair as suas características, fazer o pré-processamento<sup>1</sup> e classificá-lo em uma das classes existentes. Estes tipos de classificadores têm grande importância quer no ambiente militar quer no ambiente civil.

---

<sup>1</sup> Pré-processamento dos dados é essencialmente a redução do espaço de pesquisa, isto é, diminuição de linhas/colunas a analisar, transformar valores contínuos em discretos ou alterar idades por faixas etárias por exemplo



Figura 1: Pirâmide de Dados, Informação e Conhecimento

## 1.1 Motivação

A análise de grandes quantidades de dados para o homem seria inviável sem o auxílio de ferramentas computacionais apropriadas. Portanto, torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação. (Goldschmidt & Passos, 2005)

Os áudios são ondas mecânicas e propagam-se por vibrações das partículas do meio envolvente, dessa forma, a sua velocidade de propagação e o seu alcance depende muito da elasticidade do meio. Sendo o mar um meio elástico e mais denso que o ar, o som propaga-se com maior velocidade e alcance do que no ar (Santos C. I., 2015). Atualmente, devido à grande evolução na área de *data mining*, que se deve essencialmente ao enorme crescimento de dados existentes, tornou-se fundamental extrair destes mesmos repositórios conhecimento de forma automatizada que auxiliem o ser humano no processo de tomada de decisões.

Para presente tese, o problema de detecção e classificação dos navios através dos sons por eles produzidos constitui uma das principais tarefas dos submarinos, o operador humano a quem cabe esta função está sujeito a falhas no processo de identificação por fatores como limitações fisiológicas, condição emocional, subjetividade na interpretação e ruídos e ecos parasitas de diversas fontes que podem perturbar o sinal, portanto, uma

má identificação causa desinformação e compromete a integridade do submarino pondo em risco toda guarnição a bordo daí a necessidade de um sistema de identificação automático que o auxilie na identificação (classificação) dos navios ao seu redor para garantir a segurança e vantagem tática em relação a outras unidades navais.

## 1.2 Estrutura do Documento

A presente dissertação está composta por 8 capítulos principais começando por uma introdução ao problema e a motivação para elaborar a aplicação.

O segundo capítulo faz uma introdução à acústica submarina, pois é o meio ambiente onde é efetuada a coleta dos dados, focando-se na influência do meio envolvente bem como nos ruídos inerentes do meio ambiente e ruídos do próprio navio que, consequentemente, interferem na qualidade do som podendo levar erros de classificação ou até mesmo de detecção. Neste mesmo capítulo, é apresentada uma breve descrição de algumas das principais características da onda acústica terminando com algumas aplicações da acústica submarina para fins militares e para fins civis.

No terceiro capítulo, abordam-se o processamento de sinal e o processo de extração de características através da transformada de Fourier, da transformada de Fourier em tempo curto, Coeficientes Mel-Cepstrais e as componentes de Cromagrama, necessário para que a base de dados possua o maior número de características necessárias para a classificação de um navio.

O quarto capítulo da tese introduzem-se os conceitos básicos sobre *Data Mining* e explanam-se as etapas para o processo de tomada de decisão. No subcapítulo 4.1, referente ao pre-processamento dá-se particular foco ao processo de normalização dos dados devido à sua importância para as técnicas de previsão utilizadas. No subcapítulo 4.2, abordam-se as técnicas de agrupamento, k-Médias e Mapas Auto-Organizados, focando-se na última que poderá ser adaptado para atuar como um classificador. No último subcapítulo, falam-se as técnicas de previsão dando maior relevo à classificação por k-vizinhos, importante para a adaptação dos Mapas Auto-Organizados à função de classificador, e às

Redes Neurais Artificiais técnica mais utilizada na atualidade para a classificação de dados acústicos.

No quinto capítulo, aplica-se a extração de componentes e avalia-se a qualidade de duas técnicas de classificação referidas no capítulo anterior, a adaptação dos Mapas Auto-organizados e as Redes Neurais Artificiais, a 73 registos acústicos reais de navios, implementadas no *software* Python.

No último capítulo, são apresentadas as conclusões sobre a qualidade das duas técnicas de classificação referidas e apresentam-se sugestões para melhorar o processo de tomada de decisão, no momento da classificação de um novo registo acústico de um navio.

## 2 Acústica Submarina

A acústica submarina é a ciência que estuda a propagação de som na água e a interação das ondas mecânicas com a água bem como os seus limites. O estudo da possibilidade de os navios poderem ser detetados através da escuta do ruído por eles produzido é conhecido há muito tempo. Até mesmo o grande inventor Italiano Leonardo da Vinci no século XV foi responsável por inúmeras invenções, dentre as quais desenvolveu uma ideia de uma nave submarina, além de outros projetos para exploração aquática. Historicamente, a Acústica submarina teve os seus primeiros passos na 1ª Guerra Mundial, com o aparecimento do submarino e do SONAR. Outros acontecimentos também incentivaram ao avanço e a exploração desta ciência como o caso do naufrágio do Titanic em 1912 para criação de métodos que possibilitem a deteção de icebergs por meios acústicos. (Varela, 2001)

As ondas eletromagnéticas propagam-se de forma excelente no ar e no vácuo, mas o mesmo não acontece em meios elásticos e mais densos como o caso da água. Para este efeito, quer para comunicação, deteção e identificação usamos as ondas acústicas, da mesma forma que fazem certos animais marinhos, principalmente mamíferos que dependem do som para comunicação, proteção, localização de comida e para navegação. Os sons na água possuem melhores características de transmissão que no ar como por exemplo:

- Velocidade de propagação aproximadamente  $1500m/s$  em média;
- Podem atingir níveis sonoros muito elevados;
- Sofrem muito menos atenuação, facilitando a propagação maior distâncias.

Desta forma, as ondas acústicas hoje em dia são capazes de desempenhar o papel que as ondas eletromagnéticas desempenham na atmosfera através das aplicações no rádio e no radar para as seguintes funções, contudo com desempenhos inferiores (Varela, 2001):

- Detetar obstáculos ou alvos, função principal dos sistemas sonar, envolvendo não apenas aplicações militares como caça de submarinos e minas como também aplicações em ambiente civil como no caso das pescas, prevenção de desastres, controlo de robótica submarina e outras aplicações.
- Medição das características dos diferentes constituintes do meio, relevo do fundo, profundidade, organismos vivos, entre outras funções.

## 2.1 Influência do meio

A água do mar é um bom meio propagador do som, contudo, muitos fatores impõem limitações à propagação das ondas acústicas como por exemplo a agitação marítima, a chuva, os animais marinhos que também produzem os seus sons, uma série de fatores que criam ecos parasitas e que retardam a receção do sinal. O sinal é atenuado pela absorção da água do mar, e isso por sua vez limitará o alcance, tal como também outros fatores, como a reflexão e os múltiplos trajetos que o sinal pode seguir devido à não homogeneidade do meio.

O comportamento da velocidade de propagação ao longo da coluna de água é bastante complexo. É de conhecimento geral que a temperatura do mar à superfície varia com a posição geográfica, com a estação do ano, e com a hora do dia. Os valores de Temperatura, salinidade e pressão têm impacto na velocidade de propagação da onda por esta ser uma característica dependente da elasticidade do meio. (Pickard & Emery, 1990, pp. 94-99)

$$c = \sqrt{\frac{E}{\rho}} \quad (1)$$

- Temperatura: a variação de 1 grau Celsius é equivalente a aproximadamente uma variação de 4,5 m/s.



- A variação da temperatura é o fator dominante na variação da velocidade de propagação do som na água entre a superfície e o limite inferior da termoclina<sup>2</sup>, passando então a ser a pressão o valor dominante para as maiores profundidades.

### 2.1.1 Ruídos

O meio submarino é um ambiente acústico relativamente ruidoso por causa da grande quantidade de sons existentes e a sua facilidade de propagação. Define-se por ruído, o som sem qualquer interesse ou desagradável para o recetor podendo ser mais ou menos intenso, composto por uma só tonalidade ou por várias. A capacidade auditiva dos seres humanos é de aproximadamente 20 a 20.000Hz, sendo que essa capacidade vai diminuindo ao longo do tempo. Estes ruídos degradam a qualidade do sinal recebido por qualquer sonar, podendo induzir o software a ser utilizado a cometer erros de classificação.

Os ruídos podem ser classificados quanto à fonte que os produz por: Fontes Antropogénicas e Fontes Ambientais. Fontes Antropogénicas traduzem o ruído produzido pelos navios no mar devido à sua atividade, às atividades a bordo, às operações junto a zona costeira, o abrir e fechar de escotilhas, etc. Tudo isto são ruídos considerados de fontes antropogénica, ou seja, derivados de atividades humanas. As fontes ambientais caracterizam os ruídos naturais tais como a agitação marítima, a chuva, vagas, entre outros. Os animais marinhos e as plantas aquáticas também geram sons característicos para a sua comunicação e em alguns casos para a ecolocalização de presas. As baleias, por exemplo, emitem sons de baixa frequência capazes de se propagar a grandes distâncias entre (12 Hz a alguns milhares de Hz). (Santos C. I., 2015). Durante a ausência de sinal útil, é sentido o chamado ruído ambiente, que é o ruído do meio e o ruído produzido pela própria plataforma onde está inserido o sonar.

O ruído do tráfego marítimo é muito relevante entre 10 e 1000 Hz, com níveis dependentes de cada região. A presença de portos e de esquemas de separação de tráfego

---

<sup>2</sup> Termoclina é a camada de variação de temperatura do mar em determinada profundidade, ocorre tipicamente entre as 300 e 1000 metros abaixo da superfície

constituem um fator de variação local importante e de anisotropia espacial<sup>3</sup> da estrutura da distribuição do ruído em azimuth. Podemos categorizar três principais fontes de ruídos a bordo que são: Máquinas e Auxiliares, Hélices e o Arrastamento da água junto ao casco (ruídos hidrodinâmicos).

As Máquinas e auxiliares bem como as canalizações de vapor de água transmitem vibrações ao casco, as explosões periódicas dentro dos cilindros de um motor a *diesel*, os dentes de engrenagem, as descontinuidades das pás de uma turbina dão origem a ruídos de banda estreita e por estarem associados a um sistema de propulsão a frequência e amplitude do sinal aumentam com a velocidade. As hélices devido ao contacto direto com o mar produzem ruído distinto associado as máquinas no interior do navio, a cavitação<sup>4</sup> provocada pela rotação das pás gera ruído de bandas largas a frequências elevadas. (Varela, 2001)

Existe também o ruído causado pelos apêndices dos cascos dos navios, que causam à ré uma esteira capaz de produzir ruído, daí a necessidade de suprimir os apêndices dos submarinos. Todas as descontinuidades do casco nas obras vivas (navios de superfície) causam esse efeito e são de se evitar ao máximo.

## 2.2 Caraterísticas das Ondas Acústicas

Nas ondas acústicas sucedem-se variações no campo da pressão na água e as partículas movem-se longitudinalmente, para frente e para trás na direção da propagação da onda, causando zonas de compressão e de rarefação como se pode ver na figura 1.

---

<sup>3</sup> Anisotropia é a variabilidade ou distribuição espacial de algum elemento da natureza

<sup>4</sup> Cavitação é o fenómeno causado pela ação das hélices na água causando implosão e explosão de bolhas de ar, devido a diferença de pressão entre os dois lados das pás.

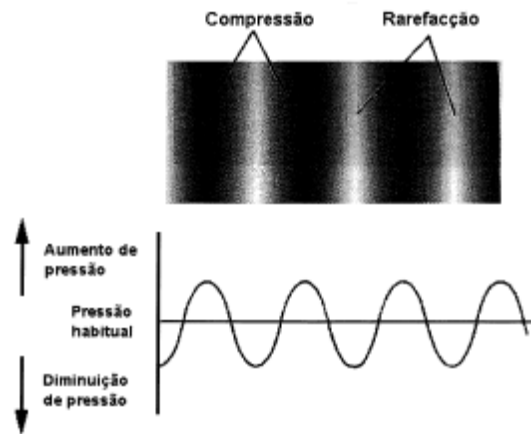


Figura 2: Representação de onda sonora ("PEEAS 43 (A)")

### 2.2.1 Comprimento de onda ( $\lambda$ ), velocidade e frequência

Durante a transmissão de informação de um ponto para o outro existe sempre um pequeno esquema de acontecimentos: Emissão  $\rightarrow$  Propagação  $\rightarrow$  Receção. As fontes produzem/emitem sinais, o meio facilita a sua propagação e por fim os recetores ou o ser humano regista a chegada do sinal.

Designa-se por comprimento de onda a distância entre dois pontos equivalentes a uma oscilação, ou seja, corresponde à distância de um ciclo em direção da propagação a que chamamos de período ( $T$ ) a uma velocidade  $c$ , e é dada pela fórmula,

$$\lambda = cT = \frac{c}{f} \quad (2)$$

Onde,

- $\lambda$  – Comprimento de onda [ $m$  ou  $cm$ ]
- $c$  – Velocidade de propagação [ $m/s$ ]
- $T$  – Período do sinal:  $T = \frac{1}{f}$  [ $1/s$ ]
- $f$  – Frequência do sinal [ $Hz$ ]

A Velocidade de propagação é a distância que a onda percorre num intervalo de tempo e é expressa em  $m/s$  no SI:

$$c = \frac{\lambda}{f} \quad (3)$$

Duas propriedades importantes a distinguir nas ondas sonoras são a amplitude e a frequência. A amplitude ( $A$ ) expressa em decibel ( $dB$ ),  $m$ ,  $cm$  ou  $Pa$  é definida como o deslocamento máximo em relação à posição de equilíbrio, dessa forma, quanto maior for a amplitude do sinal, maior será a sua intensidade ( $I$ ). A frequência é o número de vezes que se passa da mesma forma pela mesma ordenada na unidade de tempo, sendo expressa em Hertz ( $Hz$ ). Os efeitos da amplitude e da frequência no som que se ouve conseguem ser bem distintos. (Deus, Pimenta, Noronha, Peña, & Brogueira, 2014)

### 2.2.2 Intensidade Acústica

Chamamos de intensidade acústica ( $I$ ) o fluxo médio de energia por unidade de tempo através da unidade de superfície de onda, ou seja, é a quantidade média de energia na unidade de tempo numa determinada área e pode ser dada por:

$$I = \frac{P}{S} \quad (4)$$

ou

$$I = \frac{P_e^2}{\rho c} \quad (5)$$

onde,  $P$  é a potência acústica dada em Watts ( $W$ ),  $S$  é a área em  $m^2$ ,  $P_e$  é a pressão eficaz<sup>5</sup> em  $N/m^2$ ,  $\rho$  é a densidade do meio em  $kg/m^3$  e  $c$  a velocidade de propagação em  $m/s$ . A pressão eficaz por sua vez é calculada através de

---

<sup>5</sup>  $P_e$  é pressão efectiva, dada pela raiz média quadrática (*root mean square*) da amplitude da pressão acústica  $P$  (Deus, Pimenta, Noronha, Peña, & Brogueira, 2014)

$$P_e = \frac{P}{\sqrt{2}} \quad (6)$$

A Intensidade e a pressão acústica são usualmente expressas em escalas logarítmicas pela vasta gama de valores que podem assumir, escalas logarítmicas referentes a níveis de pressão e intensidade, a unidade do nível de intensidade ( $IL$ ) é designada por Bell em homenagem de Graham Bell, inventor do telefone. (Varela, 2001)

O nível de intensidade pode ser definido através de

$$IL = 10 \log_{10} \left( \frac{I}{I_{ref}} \right) \quad (7)$$

ou em alternativa, pode ser calculada por:

$$IL = 20 \log_{10} \left( \frac{P}{P_{ref}} \right) \quad (8)$$

tendo  $P_{ref}$  como a pressão de referência, que em acústica submarina é dada por  $1\mu Pa$ .

## 2.3 Aplicações Militares

O ambiente militar foi o principal impulsionador para o desenvolvimento dos sistemas acústicos desde o final da primeira guerra mundial. O sistema de deteção e identificação de objetos submarinos através de um sinal acústicos chama-se SONAR (Sound Navigation and Ranging) que é o equivalente ao RADAR (Radio Detection and Ranging) em eletromagnetismo tendo por base o princípio de funcionamento na emissão e receção de som (*ping*), podendo ser classificados em duas categorias, os Sonares Ativos e os Sonares Passivos. (Varela, 2001)

Os Sonares Ativos emitem um sinal e recolhem o eco devolvido pelo alvo. Habitualmente, num submarino em imersão, a medida do atraso do eco permite estimar a distância entre o sonar e o alvo e a receção deste numa antena adequada, permite determinar a direção angular. A respetiva análise do eco permite obter certas indicações sobre o alvo nomeadamente a sua velocidade, graças ao efeito doppler. A principal desvantagem desta

categoria de sonar é o facto de ao emitir o *ping* o submarino fica mais exposto e poderá denunciar a sua posição ao inimigo. Sonares passivos baseiam-se na deteção e estimação do som propagado pelo próprio navio, sem que haja alguma transmissão de sinal por parte do sonar. Este sistema é destinado a intercetar os sinais acústicos presentes no meio fornecendo uma vantagem tática inquestionável. (Varela, 2001)

A presente tese tem como objetivo criar um algoritmo/sistema, aplicando técnicas de *data mining* que possibilite uma melhor identificação de navios a partir da sua assinatura acústica por eles produzidos. Sistemas semelhantes a estes já existem em diversas marinhas tais como:

- A Armada Espanhola, que possui a bordo dos seus submarinos o *Sistema de Clasificación Acústica (SICLA)* (Manzoli, 2019);
- A campanha inglesa *Drumgrange*, que desenvolveu um sistema de monitorização (sistema muito usado pela *Royal Navy* do Reino Unido para controlar assinaturas acústicas de submarinos e navios de superfície – “*Platform Signature Monitoring System*”, 2014) (Santos C. I., 2015);
- O Brasil, por sua vez, utiliza o Sistema de Deteção, Acompanhamento e Classificação dos Contatos (SDAC), sistema desenvolvido pelo grupo de Sonar do Instituto de Pesquisas da Marinha (IPqM) e tem como principal objetivo fornecer apoio ao operador sonar. (Instituto de Pesquisas da Marinha do Brasil, 2019)

## 2.4 Aplicações Cíveis

A acústica submarina no ambiente civil é um sector de atividade industrial e científico devido a necessidade de instrumentação dos programas de vigilância do ambiente bem como de desenvolvimento de engenharia em offshore e de pesca industrial, e citaremos a seguir alguns exemplos das aplicações da acústica marítima no âmbito civil.

*“Sondadores Batimétricos, eis que é um sonar especializado na medição da altura da água emitindo verticalmente em direção ao fundo o sinal num feixe estreito e mede o retardo do eco retornado pelo fundo”* (Varela, 2001)

A tecnologia acústica tem sido uma grande força impulsionadora no desenvolvimento das técnicas de pescas, hoje em dia os navios dependem quase totalmente do sonar acústico e de sirenes para detetar cardumes. Esses tipos de sonares direcionados para a pesca são os seguintes:

- *CHIRP (Compressed high intensity Radar Pulse)*, considerada o maior avanço tecnológico desde a criação dos *Fish Finders*. A sua inovação, ao contrário das sondas convencionais que apenas utilizam o sinal com uma só frequência, é a de utilizar múltiplas frequências, resultando numa capacidade de detetar peixe a distâncias muito maiores que um sonar convencional, acrescentando que podem distinguir objetos muito próximos um do outro. (CHIRP Sonar Technology, 2019);
- Sondas Laterais, que permitem observações de grande precisão do que acontece ao redor do navio até profundidades de 200 m. Para que apresente uma clareza necessária, a imagem é monocromática. Sistemas também utilizados em geologia marítima e em cartografia;
- *Garmin*, com a aplicação “*MyCast*” da *Garmin* disponível na plataforma *Android*, que fornece ao utilizador informações atualizadas sobre os melhores pontos e horas para a pesca e, ainda, informações meteorológicas e de marés. (Pesca e Companhia, 2019);
- *DPS (Digital Signal Processing)*, que utiliza métodos de inteligência artificial para distinguir os dados de interesse daqueles que não têm interesse como, por exemplo, cardumes e peixes grandes das estruturas submersas, filtrando os ruídos. (Pesca e Companhia, 2019).





### 3 Processamento de Sinal

O processamento digital de sinal consiste na análise e/ou modificação (sequência discreta de números) de sinais por forma a extrair informações ou torná-los mais apropriados para determinadas aplicações. (Moura J. , 2009)

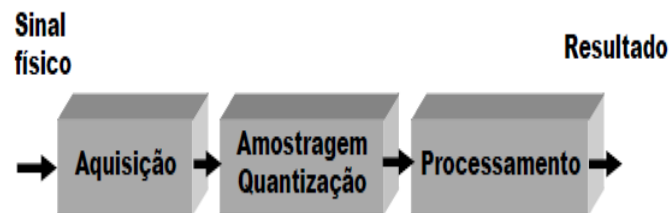


Figura 3: Diagrama Simplificado do DSP (Tan & Jeang, 2013)

Os sinais são definidos como padrões de variações que representam informação, podendo ser grandezas físicas que variam no tempo, espaço ou em qualquer outra variável independente. Para o seu processamento é necessário convertê-lo de sinal analógico para sinal digital.

O teorema da amostragem define que o sinal contínuo, para que possa ser devidamente representado, deve obedecer alguns parâmetros. Em primeiro lugar, o sinal original deve ser coletado a uma frequência definida pelo utilizador ( $f_a$ ), Frequência de amostragem, que deverá ser no mínimo superior ao dobro da maior frequência ( $f_{max}$ ) Contida no sinal original (Santos C. I., 2015, p. 13):

$$f_a > 2 \times f_{max} \quad (9)$$

Portanto, o sinal analógico passará por ser representado por um conjunto de números discretos em instantes devidamente espaçados. A esse espaço chamamos de Intervalo de amostragem ( $T_a$  ou  $\nabla T$ ), que convém que seja amostrado em tamanho suficiente para que se possa extrair a componente de maior frequência.

Na figura 3, apresenta-se um sinal analógico (contínuo), as suas amostras digitais e intervalo de amostragem.

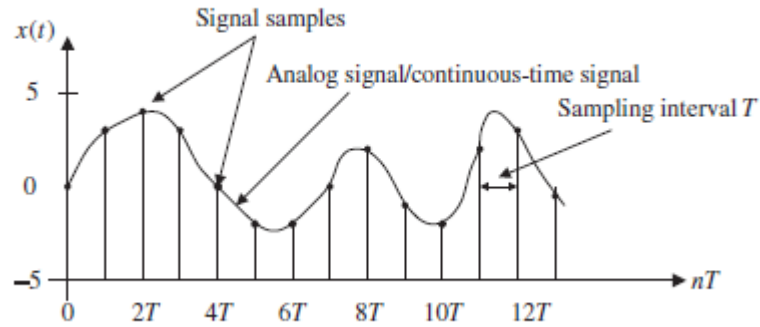


Figura 4: Sinal Analógico e os respectivas amostras digitais e respetivo Intervalo de Amostragem (Tan & Jeang, 2013, p. 16)

Durante o processo de amostragem, a frequência mínima admissível chama-se Frequência de Nyquist ( $f_n$ ) e é dada por:

$$f_n = \frac{1}{2 \times T} = 2 \times f_{max} \quad (10)$$

Ou seja, para retirar uma amostra de um sinal áudio com frequência de até  $4\text{ kHz}$ , a taxa ou frequência de amostragem mínima é de pelo menos  $8\text{ kHz}$ , ou 8000 amostras do áudio por segundo, que será a frequência de Nyquist. A figura abaixo representa um sinal cujo o teorema da amostragem foi respeitado e um segundo sinal que não cumpriu com o mesmo teorema, tratando-se de um sinal com a intervalo de amostragem de  $(T) 0.01\text{ s}$  e uma frequência de amostragem de  $100\text{ Hz}$ . O primeiro gráfico representa um sinal com  $40\text{ Hz}$  de frequência que claramente respeitou o teorema de amostragem na fórmula (9) sendo que,  $80 < 100$ . Contudo, no segundo gráfico temos um sinal com  $90\text{ Hz}$  de frequência que é subamostrado pois não respeitou a regra do teorema de amostragem,  $2f_{max} = 180 > f_a$ . (Tan & Jeang, 2013)

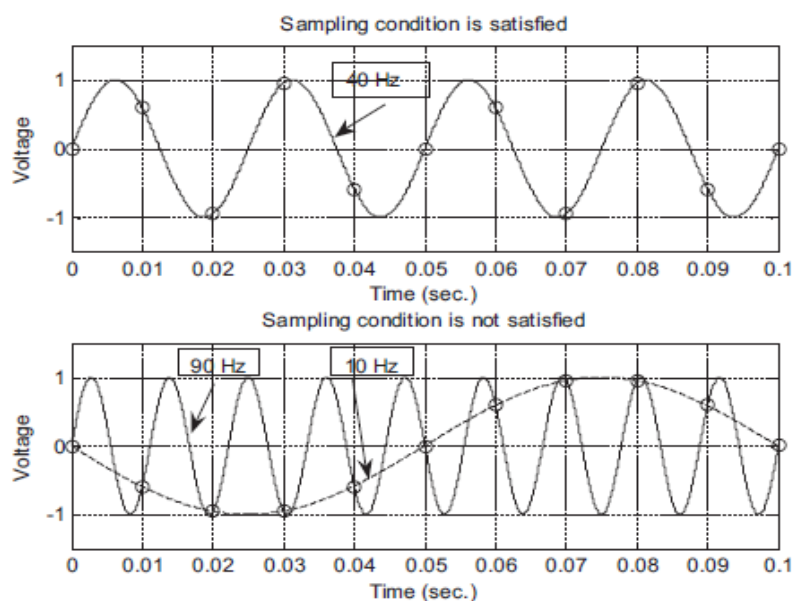


Figura 5: Sinal bem amostrado vs Sinal Mal Amostrado. (Tan & Jeang, 2013, p. 17)

Podemos reparar que, no segundo gráfico, com base nas amplitudes da amostra do sinal, não conseguimos dizer se o sinal amostrado vem da amostragem de uma onda senoidal de 90 *Hz* (linha contínua) ou se provém de uma onda senoidal de 10 *Hz* (linha tracejada) elas não são distinguíveis e pelo sinal amostrado ser mesmo de um sinal de 90 *Hz* chamamos o sinal de 10 *Hz* de Ruído *Aliasing*. (Tan & Jeang, 2013)

Este capítulo tem particular relevância na presente dissertação na fase de extração de recursos, a extração de recursos consiste em transformar o conjunto de dados iniciais (sons) em dados que podem ser representativos e usados pelo classificador, é um problema diferente relativamente a escolha de recursos que consiste em selecionar os componentes mais relevantes do conjunto dos recursos extraídos para a classificação final.

Na aplicação que se pretende desenvolver, iremos utilizar técnicas que permitam a extração de características/componentes, tais como transformadas de Fourier de tempo curto, coeficientes cepstrais da frequência Mel, extração de componentes dos respetivos cromagramas, dos espectrogramas em escala de Mel, do contraste espectral e do Tonnetz. Nos próximos sub-capítulos, introduz-se uma breve noção de cada uma das técnicas referidas.

### 3.1 Transformadas de Fourier

Jean Baptiste Joseph Fourier matemático e físico francês, no século *XIX*, foi um dos pioneiros dos estudos sobre dissipação do calor e demonstrou que uma onda periódica pode ser representada por uma soma de ondas senoidais (séries de Fourier), foi o primeiro a aplicar em problemas de dissipação e transferência de calor esta técnica de decomposição sinusoidal. Esta tem vindo a ser amplamente usada nas Ciências e nas Engenharias desde então. Joseph Fourier defende que qualquer série de comprimento finito, repetida infinitamente, definida num intervalo  $[0, T]$ , pode ser reproduzida através de uma soma linear de cossenos e senos, pois trata-se, na verdade, de uma distribuição complexa que integra informação relativas às sinusoides que compõe o sinal  $x(t)$ , o que possibilita a representação do sinal original em domínios diferentes. (Santos C. I., 2015)

A representação do sinal no domínio da frequência é chamada de Espectro do Sinal, se as fases das funções sinusoidais são ignoradas e usarmos apenas as suas amplitudes esta representação será invariante no tempo atingindo assim um dos objetivos usuais da extração de recursos/características. O quadrado do módulo do Espectro é chamado Densidade ou Densidade espectral de energia, para sinais de energia. A sua utilidade é imensa tendo em conta que permite determinar o poder do sinal original (Energia) e qualquer banda de frequências.

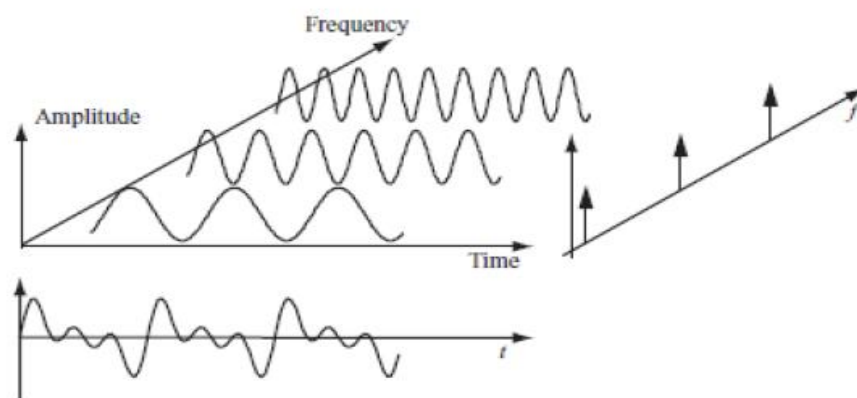


Figura 6: Representação de um sinal, decomposto em três sinais com diferentes frequências e a mesma amplitude, vistos no domínio do tempo e da frequência (Brandt, 2011, p. 170).

A transformada de Fourier (TF),  $X(\omega)$ , de um sinal  $x(t)$ , por definição, é dada por

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-i\omega t} dt, \quad (11)$$

onde  $\omega$  é a frequência angular dada por  $\omega = \frac{2\pi}{T}$  e  $i$  é a unidade dos números complexos puros.

A partir de (11) vemos a TF na sua forma contínua, o que torna inviável para o cálculo computacional visto que só é possível efetuarlo para um conjunto de valores e não para a sua totalidade, portanto, para a análise de dados é necessário definir uma função discreta que se aproxime da função contínua. Uma técnica muito utilizada em áreas como a análise espectral, acústica, geração de imagens e sistemas de comunicação, é a que transforma as amostras do sinal no domínio do tempo para os componentes do domínio da frequência, conhecida como o *Discrete Fourier Transform* (DFT) definida através de

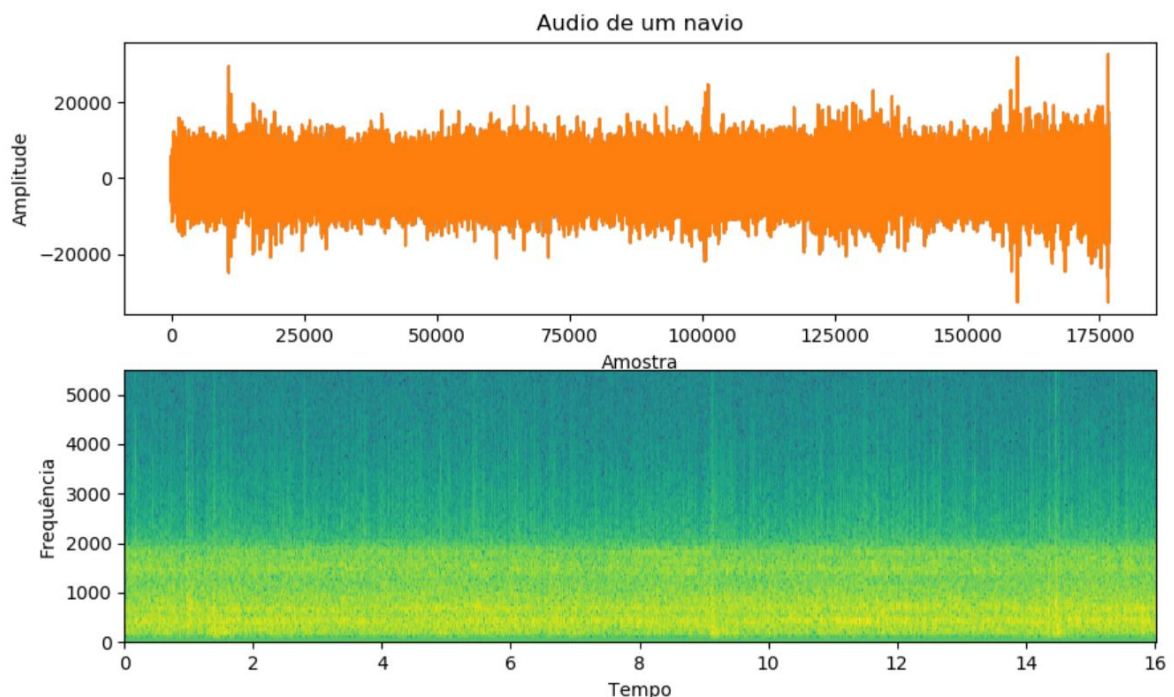
$$X(m) = \sum_{n=0}^N x(n)e^{-i2\pi nm/N} \quad (12)$$

onde  $X(m)$  será o valor discreto da DFT,  $m$  será um valor inteiro entre 0 e  $N - 1$  que corresponderão às frequências angulares entre 0 e  $2\pi$  e  $x(n)$  será uma sequência de valores de entrada no domínio do tempo para  $n$  entre 0 e  $N - 1$ , para  $N$  o número de amostras da sequência de entrada e o número de pontos no domínio da frequência da DFT de saída. (Santos C. I., 2015)

## 3.2 Transformada de Fourier em Tempo curto

A transformada de Fourier em tempo curto (STFT) é uma ferramenta poderosa para análise e processamento de sinais de áudio e uma forma inteligente do uso das DFT. O objetivo da STFT de forma geral é de dividir o sinal em quadros possivelmente sobrepostos usando uma técnica de janela móvel e calcular a DFT em cada quadro ou janela. O comprimento da janela (subconjunto de dados com  $N \leq M$  pontos) tem grande influência na análise do sinal pois, enquanto as maiores janelas levam a uma melhor resolução de frequência, as menores fornecem informação detalhada no domínio do tempo.

A sua representação é feita através de uma imagem chamada Espectrograma do Sinal e que permite visualizar a evolução do sinal nos domínios de tempo e de frequência. Usa-se a Magnitude ou a Magnitude Quadrada dos coeficientes STFT em escala linear ou logarítmica ( $dB$ ). A forma usual do espectrograma é planar, tendo diferentes cores para indicar a intensidade da densidade de energia, variando do violeta ao vermelho do espec-



tro visível.

Figura 7: Audio e Espectrograma de um segmento de um registro sonar de um navio.

(implementação em *Python*<sup>TM</sup>)

### 3.3 Coeficientes Mel-Cepstrais

*Mel-frequency cepstral coefficients* (MFCC) são os recursos mais utilizados em aplicativos de reconhecimento de fala, reconhecimento de pessoas a partir das suas vozes, sendo na sua natureza um problema similar ao tema desta tese. Portanto, na fase de extração de componentes, após extrair o STFT, calculamos os MFCC. Para que se efetue a extração desses coeficientes MFCC's deveremos proceder como está apresentado na Figura 7.

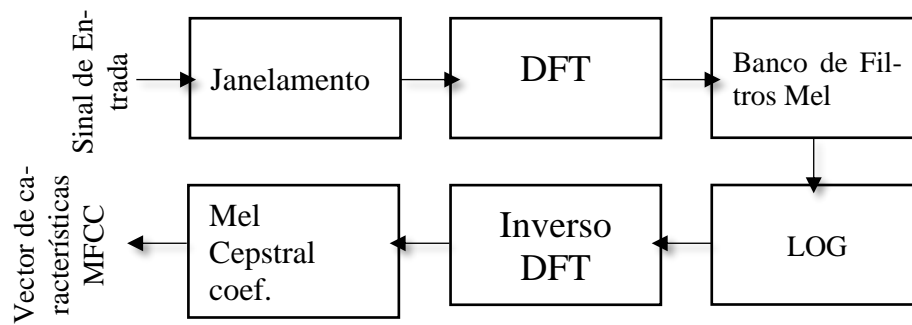


Figura 8: Passo para a extração do vetor MFCC's (Majeed, Husain, & Samad, 2015)

Este método foi concebido por Bridle e Brown (Bridle & Brown, 1974) e logo a seguir desenvolvido por Davis & Mermelstein (Davis & Mermelstein, 1980) com objetivo de extrair um vetor de características com toda a informação linguística de um sinal de voz utilizando a escala Mel para analisar as diferentes frequências presentes no espectro. A escala de Mel foi desenvolvida por Stevens e Volkmann por meio de experiências com o objetivo de identificar a forma como as diferentes frequências eram interpretadas pelo aparelho auditivo humano, ou seja, para descrever uma relação entre a frequência real e o que era interpretado. Concluiu-se que a relação é linear de 0 a 1000 Hz e que a partir de frequências superiores a 1000 Hz a relação pode ser descrita de forma logarítmica, ou seja, da forma

$$F_{mel} = \frac{1000}{\log_{10}(2)} \times \left[ 1 + \frac{F_{Hz}}{1000} \right] \quad (14)$$

onde  $F_{mel}$  é a frequência na escala de mel medida em mel e  $F_{Hz}$  é a frequência normal medida em Hz. (Davis & Mermelstein, 1980)

### 3.4 Componentes de Cromagrama

Os atributos como o Croma (*chroma*), sonoridade (*loudness*) e timbre, são atributos que estão relacionados com a percepção humana do som e são muito usados em algoritmos de identificação de músicas. A sonoridade é uma sensação subjetiva relacionada com as variações da pressão acústica, sendo também influenciada pela frequência e pela duração do sinal, observando o facto de que os sons mais curtos geram uma menor percepção de sonoridade. (Távora, 2017)

O *pitch* espectral ou altura é a sensação subjetiva que cada indivíduo possui em relação à frequência de cada Tom, sendo também a característica que determina se um som é alto ou baixo, estando relacionado com a frequência e com a duração do sinal, porém, restrito à interpretação individual do ouvinte. Um som com uma única frequência é denominado Tom e a interligação da frequência física de um tom com a nossa sensação denomina-se *pitch*, ou seja, para cada mudança na frequência física implicará uma mudança semelhante no *pitch*. (Carneiro, 2019)

O conceito de timbre é aplicado ao modo de como um instrumento soa, sendo o que nos auxilia a diferenciar um instrumento musical de outro, tal e qual o timbre, que nos fornece capacidades de diferenciar a pessoa que está falar de outra, qual o objeto que caiu e qual o instrumento musical a tocar, sendo, para tal, definido como a propriedade do som que caracteriza a sua fonte. (Manzoli, 2019)

O Croma (*chroma*) é um atributo associado ao *pitch*, o espectro é dividido em oitavas e em cada oitava existem 12 sub-bandas que definem as classes de pitch. As classes de *pitch* equivalentes, mesmo pertencendo a oitavas diferentes, produzem uma sensação auditiva semelhante. Portanto, o croma será útil na descrição perceptual dos sons e o cromagrama representa a variação temporal da energia espectral de cada uma das classes de *pitch*, mapeando todo espectro em uma única oitava. (Távora, 2017)



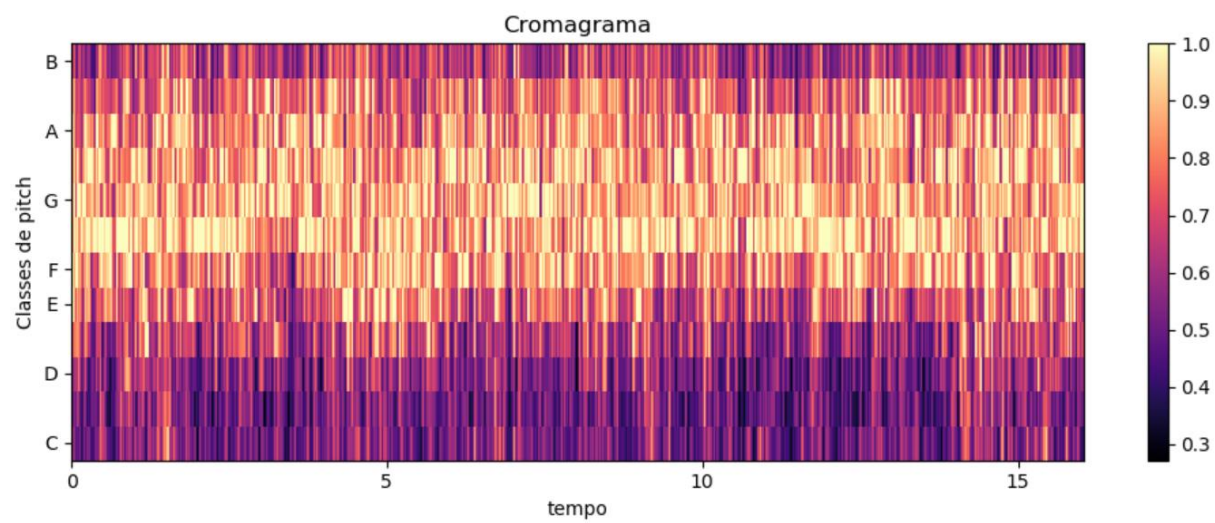


Figura 9: Cromagrama de um Sinal de um registro de um navio. (Python™)



## 4 *Data Mining*

Atualmente, com o avanço tecnológico, a grande evolução dos dados, as constantes transações comerciais e com o surgimento da internet, as organizações são obrigadas a acumular grandes quantidades de dados, o que torna quase impossível, para o ser humano, verificar e analisar toda a base de dados, sendo para tal necessário criar ferramentas capazes de transformar esse conjunto de dados em informação útil ou conhecimento. O *Data Mining* (DM), por ser uma área muito ampla, possui diversas definições associadas a este conceito, contudo, segundo (Han & Kamber, 2012), DM é definida como sendo “o processo de descoberta de padrões interessantes em grandes blocos de dados, armazenados em base de dados, *data warehouse*<sup>6</sup> ou outros repositórios de informação”. Portanto, podemos afirmar que o DM define o processo de captura e análise de um grande conjunto de dados de forma a extrair algum significado, sendo usado para descrever características do passado ou para prever tendências futuras. O DM é a componente cognitiva das organizações e possibilita a extração de conhecimento a partir da experiência adquirida, a previsão de acontecimentos, a identificação de situações e a otimização de processos. Surgiu a partir de uma interligação da Estatística Clássica, da Inteligência Artificial (IA) e Aprendizagem de Máquina (*Machine Learning*), e envolve a aplicação de técnicas associadas a essas áreas de modo a permitir resolver problemas de predição e de aquisição de novo conhecimento. (Carvalho, 2014)

A aquisição de novo conhecimento é um processo conhecido por KDD (*Knowledge Discovery in Database*) e pode-se afirmar que o DM é uma das etapas do processo de KDD. Na figura 9, podemos observar as diferentes etapas do processo de KDD, sendo importante referir que estas etapas nem sempre são executadas de forma sequencial. (Carvalho, 2014, pp. 23-25)

O primeiro passo durante o processo de KDD é o de compreender o objetivo pretendido para o atingir após a organização dos dados que constituirão o histórico, sendo

---

<sup>6</sup> *Data warehouse* é um repositório com múltiplos dados unificados sob um único esquema com objetivo de facilitar a sua gestão. Ex: Uma empresa que tenha várias espalhadas por um país cada uma com a sua base de dados, unir todas bases de dados num único esquema, será o processo de *data warehousing*.

para tal importante fazer um pré-processamento inicial dos dados. O pré-processamento é uma área ampla e considerada a tarefa mais demorada no processo KDD e de grande importância, pois trata da qualidade dos dados, sendo essencial para se obter resultados confiáveis.

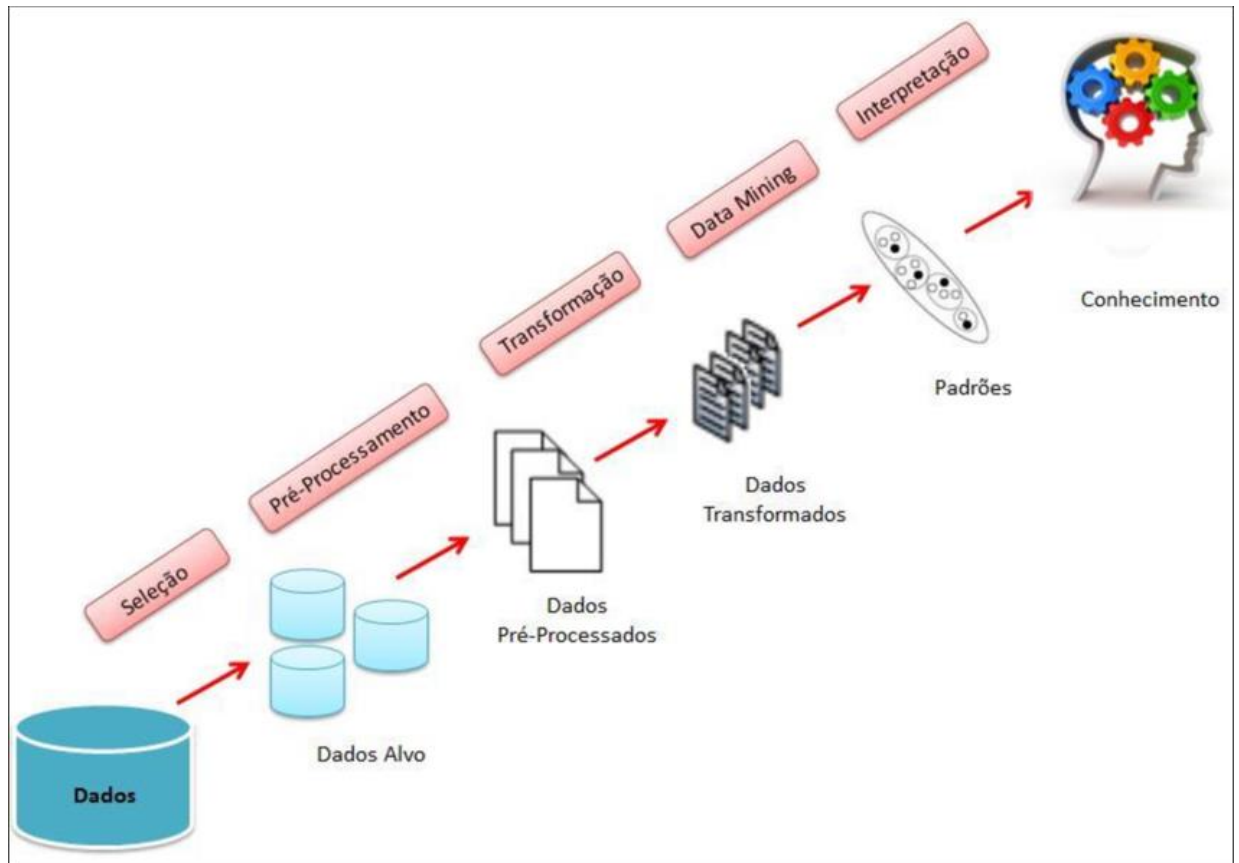


Figura 10: Etapas do Processo de KDD. (Camargo, 2016)

A etapa da seleção é a responsável por limitar o espaço de pesquisa, eliminando atributos<sup>7</sup> que não têm qualquer interesse no processo contrastando com a etapa de pré-processamento onde os dados obtidos são agrupados ou transformados de modo a tornar a pesquisa mais eficiente, reduzindo, por vezes, o número de linhas (registos, instâncias ou objetos) ou colunas (atributos, variáveis ou componentes) a analisar pode ser feita a partir da transformação de atributos com valores contínuos em atributos com valores discreto, por

---

<sup>7</sup> Atributo é uma propriedade ou característica de um objeto que pode variar de um objeto ao outro. (Tan, Steinbach, & Kumar, 2005)

exemplo substituir idades por faixas etárias ou generalizando os atributos como regiões o que permite com o que os dados sejam agrupados e analisados ao nível de freguesias (Santos & Ramos, 2017). De outro modo, podemos representar a figura 9 da seguinte forma:

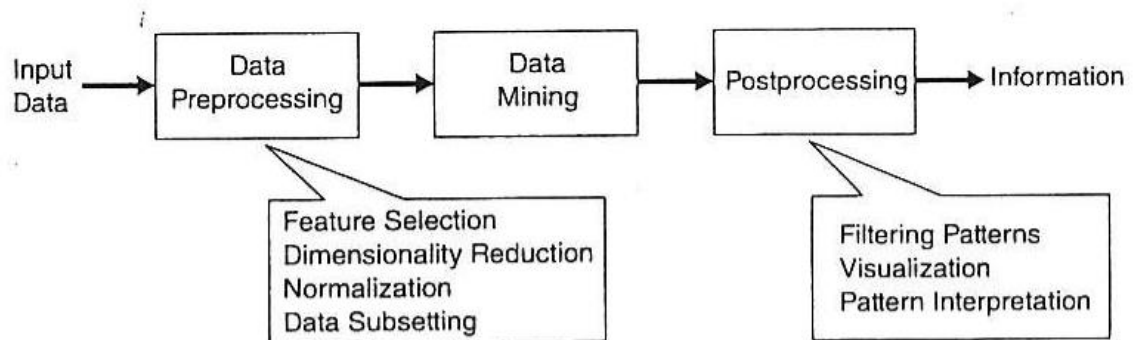


Figura 11: Passos para o processo de KDD (Simplificado). (Tan, Steinbach, & Kumar, 2005)

Como podemos observar na figura 10, outros autores (Tan, Steinbach, & Kumar, 2005) consideram pré-processamento tudo aquilo que antecede a aplicação de técnicas de *Data Mining* e pós-processamento tudo o que sucede. As variadas técnicas de *data mining* usadas para dar solução aos problemas podem ser descritas como técnicas de: Descrição/Visualização dos Dados, Estimação, Classificação, Clusterização e problemas de Associação sendo cada um deles devidamente esclarecido na bibliografia de (Larose, 2005).

## 4.1 Pré-processamento dos dados

Como já foi referido, antes de se aplicar as diferentes técnicas de MD nos dados será necessário prepará-los com o intuito de diminuir o esforço computacional (facilitando o processo posterior) e aumentar a eficiência dos modelos aplicados. Todo conhecimento à priori acerca do problema será induzido nesta etapa, bem como serão minimizados ao máximo os problemas de coleta e medição dos dados, pois poderão ser causados por erro humano ou mesmo por limitação do aparelho de medida ou coleta, podemos encontrar também nos dados, ruídos, *outliers* e valores omissos ou duplicados. O pré-

processamento dos dados pode compreender de entre outras qualquer uma das seguintes etapas (Zhang, 2003).

- Limpeza dos dados: detecção e correção dos dados, efetuando a sua substituição, alteração ou a sua eliminação.
- Redução dos dados: conceptualmente resume-se à redução da sua quantidade, preservando os dados e variáveis que sejam fulcrais para o processamento.
- Transformação dos dados: sucintamente resume-se à transformação dos dados brutos em dados com formatos mais apropriados para o processamento posterior.

Tendo em conta o objetivo do presente trabalho, será aprofundado o processo de transformação de variáveis associado à técnica da normalização dos dados. Este pré-processamento é utilizado quando se pretende que algumas variáveis da nossa base de dados possuam a mesma ordem de grandeza.

#### 4.1.1 Normalização Min-Max

O procedimento de normalização consiste em impor a mesma escala a todos os valores dos atributos/variáveis por forma a ficarem todos no mesmo intervalo específico por exemplo  $[0; 1]$  ou  $[-1; 1]$ , ou, ajustar todos os atributos à distribuição normal. É especialmente útil quando se pretende usar algoritmos que se baseiam-se em distâncias e para acelerar o processo de “convergência” de um algoritmo de *Machine Learning* (Redes Neurais Artificiais por exemplo).

A normalização Min-Max consiste em transformar a variável  $Y$  numa nova variável normalizada  $Y'$  através de

$$Y' = \frac{Y - \min(Y)}{\max(Y) - \min(Y)} \quad (15)$$

onde  $\min(.)$  e  $\max(.)$ , representam as o mínimo e o máximo de todos os valores, respetivamente.

Esta transformação tem a vantagem de preservar a relação dos valores originais de cada atributo, sendo ao mesmo tempo uma das normalizações mais simples de aplicar e, portanto, sendo das mais comuns. (Clodoaldo A. M. Lima, 2015)

#### 4.1.2 Normalização Standard

A normalização *standard* ou Padrão é definida a partir do valor médio ( $\mu$ ) e desvio padrão ( $\sigma$ ). No entanto, por serem desconhecidas, recorre-se à média e desvio padrão amostrais, sucessivamente, dos valores existentes nos atributos. É particularmente útil quando valores como o mínimo e o máximo não são conhecidos por haverem *outliers* no conjunto de valores. O objetivo é o de aproximar a uma distribuição normal o conjunto de valores, assumindo que a variável  $X$  tem distribuição normal, representando-se simbolicamente por  $X \sim N(\mu, \sigma)$  que apresenta a célebre distribuição em forma de ‘sino’ (Figura 12) . (Duarte & Pereira, 1990)

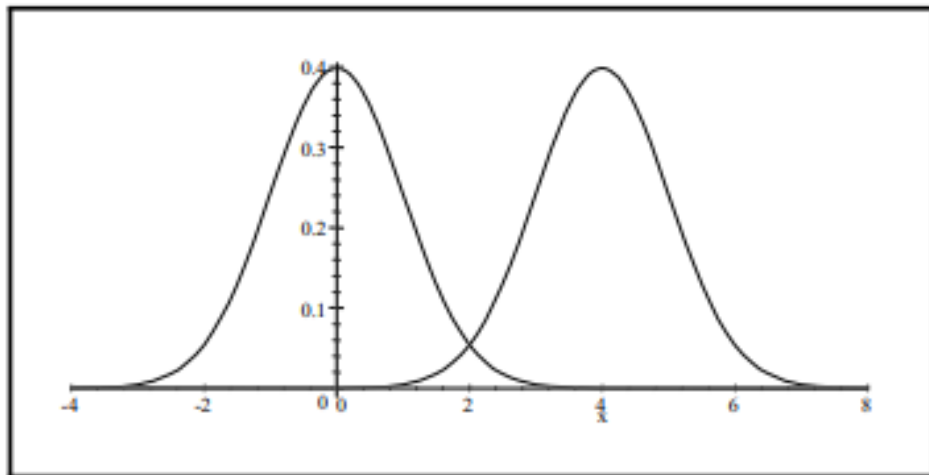


Figura 12: Função de Densidade da Normal com  $N(0,1)$  e  $N(4,1)$ . (Duarte & Pereira, 1990)

Se a variável  $X$  é uma variável aleatória com distribuição normal  $Y \sim N(\mu, \sigma)$  a variável  $Z$ , ou, o valor padronizado será dado por

$$Z = \frac{Y - \mu}{\sigma}. \quad (16)$$

Em qualquer caso, a distribuição da nova variável será distribuída normalmente com valor médio 0 e desvio padrão 1, ou seja,  $Z \sim N(0,1)$ . (Duarte & Pereira, 1990) No entanto, na prática, em vez de se recorrer a  $\mu$  e a  $\sigma$ , recorre-se, respetivamente às suas estimativas amostrais,  $\bar{x}$  e  $s$ .

## 4.2 Técnicas de Agrupamento

Fazendo parte do tipo de aprendizagem não supervisionada, as técnicas de agrupamento ou de clusterização de registos, observações ou casos, em classes de objetos semelhantes consistem em criar subgrupos (*cluster*) de dados com várias semelhanças entre si. O objetivo principal das técnicas de agrupamento são as de segmentar todo o conjunto de dados em subgrupos relativamente homogéneos onde a similaridade dentro do mesmo *Clusters* é maximizada e a similaridade dos registos fora do cluster é minimizada (Larose, 2005) outros autores tais como (Hastie, Tibshiran, & Friedman, 2008, pp. 520-522) afirmam que o objetivo principal da análise de *clusters* é a de fornecer a noção de similaridade (ou dissimilaridade) entre os objetos individuais que estão sendo agrupados. Na figura abaixo, observamos a representação de alguns dados simulados agrupados em três grupos (laranja, azul e verde) por meio da técnica mais popular, o algoritmo de k-médias (*k-means*). (Hastie, Tibshiran, & Friedman, 2008) Mais adiante iremos descrever mais ao pormenor o algoritmo de K-médias.



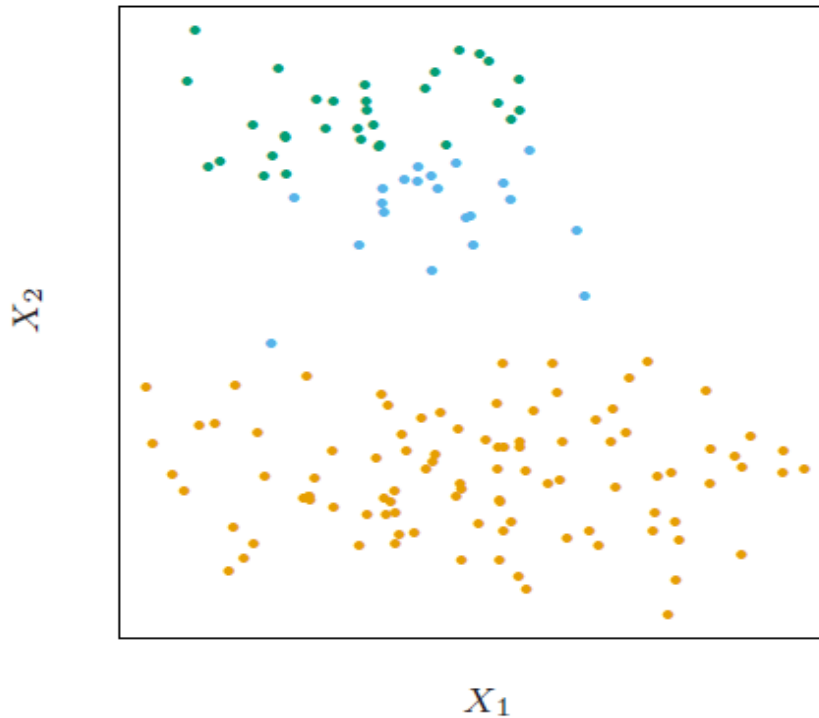


Figura 13: Dados simulados no plano agrupados em 3 classes pelo algoritmo k-médias.  
(Hastie, Tibshiran, & Friedman, 2008, p. 521)

Muitas vezes a análise de *clusters* é chamada de segmentação de dados em alguns aplicativos e é usado muitas vezes para detecção de *outliers* (valores isolados de qualquer *cluster*), quando, os *outliers* têm maior interesse que os casos comuns como por exemplo em aplicações para detecção de fraude com cartões de crédito (exemplo compras não frequentes, muito caras e raras com determinado cartão podem ser do interesse e indicadas como possíveis fraudes).

Nas técnicas de agrupamento não existem variáveis objetivo para o agrupamento por *Cluster*, e a sua tarefa principal não é a de classificar, no entanto, pode-se usar como classificador pois no processo de agrupamento estaremos a criar grupos/classes com similitudes que por sua vez poderão corresponder a uma classificação comum, dessa forma, na presença de um novo dado, a sua localização num certo *cluster* poder-se-á atribuir uma classe que seja comum aos elementos desse *cluster*.

### 4.2.1 K-médias

O agrupamento de dados está em forte desenvolvimento, contribuindo em áreas de pesquisa como o *Data Mining*, o *Machine Learning*, as tecnologias de bancos de dados espaciais, a recuperação de informações entre outros, etc. O foco principal na análise em *cluster* está baseado em distâncias (Han & Kamber, 2012, p. 482). O grau de semelhança ou proximidade entre os dados determina-se com recurso a uma função de distância  $d(x, y)$ , com as seguintes propriedades (Larose, 2005):

1.  $d(x, y) \geq 0$  e  $d(y, x) = 0$  se e só se  $x = y$ ;
  2.  $d(x, y) = d(y, x)$ ;
  3.  $d(x, z) \leq d(x, y) + d(y, z)$ ;
- (17)

Algumas das distâncias utilizadas são as distâncias de *Minkowski*, em particular a de *Manhattan*<sup>8</sup> e a *Euclidiana*<sup>9</sup>

- $d_{Minkowski}(x, y) = \sqrt[p]{\sum_{i=1}^n |x_i - y_i|^p}$

(18)

- $d_{Manhattan}(x, y) = \sum_{i=1}^n |x_i - y_i|$

(19)

- $d_{Euclidiana}(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2}$

(20)

O algoritmo de clusterização por *k*-médias é um algoritmo simples e eficaz para encontrar *clusters* em dados. O algoritmo segue os seguintes passos (Larose, 2005, p. 153):

- Passo 1: Definir quantos *clusters*,  $k$ , que se pretendem.
- Passo 2: Atribuir aleatoriamente  $k$  dados como centros/centroides iniciais do *cluster*.

---

<sup>8</sup> Distância de *Minkowski* de ordem 1

<sup>9</sup> Distância *Euclidiana* de ordem 2

- Passo 3: Para cada dado da base de dados, localizar o centroide mais próximo. Esse dado será agrupado no *cluster* associado a esse centroide.
- Passo 4: Para cada um dos  $k$  *clusters*, calcule o novo centroide de cada *cluster* e atualize as suas coordenadas.
- Passo 5: repita os passos 3 a 5 até à convergência.

O critério “mais próximo” na etapa 3 é geralmente efetuado a partir da aplicação direta da distância Euclidiana onde se escolhe a ligação onde esta é mínima, embora outros critérios também possam ser aplicados. O centroide na etapa 4 é encontrado da seguinte forma:

Considerando que existem  $N$  pontos de dados  $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots (a_n, b_n, c_n)$  no *cluster*, o centroide desse *cluster* será o centro de gravidade desses pontos e está localizado nas coordenadas  $\left(\frac{\sum a_i}{n}, \frac{\sum b_i}{n}, \frac{\sum c_i}{n}\right)$ .

Existem muitas variantes do procedimento básico de K-médias. Alguns produzem um agrupamento hierárquico aplicando o algoritmo, com  $k = 2$ , ao conjunto de dados geral e depois repetindo, recursivamente, dentro de cada *Cluster* o mesmo processo. A escolha do número de *clusters*  $k$  é uma grande incógnita, pois, raramente se tem a noção de qual a informação contida nos *clusters*, quanto mais o seu número, sendo esse o verdadeiro objetivo do *Clustering*, descobrir. Um modo eficaz é tentar diferentes valores para  $k$  e escolher o melhor a partir de uma avaliação do conteúdo de cada cluster. (Witten & Frank, 2005, p. 139)

#### 4.2.2 Mapas auto-organizados (SOM)

Os mapas auto-organizados de *Kohonen* (SOM) são uma técnica de agrupamento de rede neural não supervisionada e foram introduzidos em 1982 por *Tuevo Kohonen* (Kohonen, 1982). Embora inicialmente aplicada à análise de imagem e de som, os mapas *Kohonen* são, no entanto, um mecanismo eficaz para análise de agrupamentos cujo o objetivo é converter um sinal de entrada complexo de alta dimensão em um mapa discreto de baixa dimensão mais simples. Assim, os SOM's são bastante apropriados para a análise de *clusters*, onde os padrões ocultos subjacentes entre os registos e campos são procurados (Larose, 2005). O próprio *Kohonen* descreve o SOM como uma “ferramenta de visualização e análise de dados de alta dimensionalidade”, no entanto esta é também usada para agrupamento de dados, redução de dimensionalidade, classificação e mineração de dados (Lobo V. J., 2002, p. 38).

Os SOM's também são um tipo de rede neuronal (abordado na secção 4.3.3), mas, ao contrário das redes neuronais artificiais mais comuns os SOM's não possuem qualquer camada oculta, os dados da camada de entrada passam diretamente para camada de saída, que por sua vez é representado na forma de treliça<sup>10</sup> de uma ou duas dimensões (Larose, 2005, p. 164). Os SOM's estruturam os nós de saída em *clusters* de nós, onde os nós mais próximos são mais semelhantes entre si do que os outros nós que estão mais distantes. Estes são baseados numa aprendizagem competitiva, onde os nós de saída competem entre si para serem o nó vencedor (ou neurónio vencedor). Do mesmo modo, podendo ser visto como uma versão restrita do *clustering k-means* onde os protótipos são encorajados ou forçados a ficar numa variedade unidimensional ou bidimensional no espaço dos dados. A distribuição resultante também é referida como um mapa topológico restrito, uma vez que as observações originais de alta dimensão podem ser mapeadas nos sistemas de coordenadas bidimensional. (Kohonen, 2013)

---

<sup>10</sup> Sistema de vigas cruzadas usado no travejamento de pontes e telhados (ex.: treliça de madeira; treliça metálica).

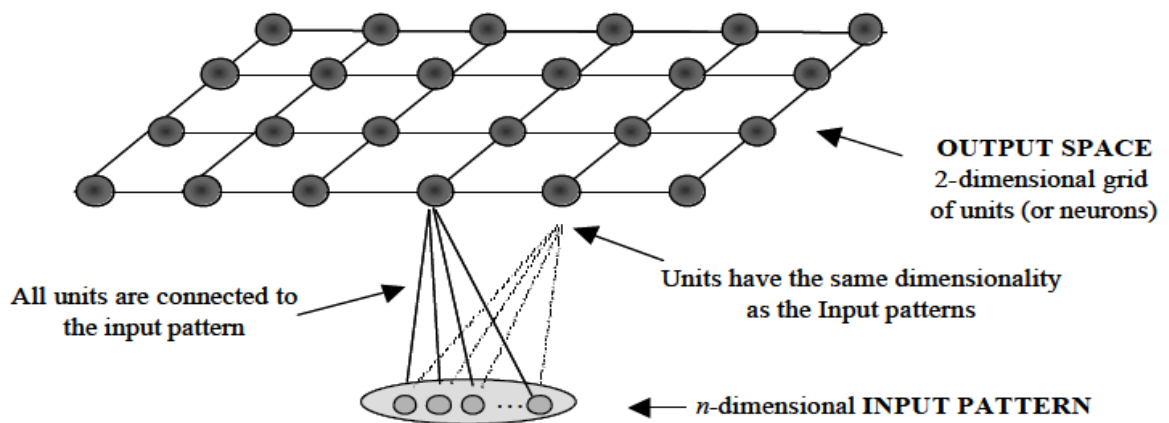


Figura 14: Estrutura Básica de um Mapa Auto-organizável (SOM). (Lobo V. J., 2002)

Considera-se um SOM como uma rede retangular, onde cada neurónio se liga a outros quatro, (ou hexagonal, onde cada neurónio possui seis ligações a outros neurónios) como ilustra a figura 14 onde cada um dos  $k$  protótipos  $w_j \in \mathbb{R}^p, j = 1, \dots, k$ , são parametrizados em relação a um par de coordenadas inteiras  $l_j \in Q_1 \times Q_2$  onde  $Q_1 = \{1, 2, \dots, q_1\}$  e  $Q_2 = \{1, 2, \dots, q_2\}$ , e sendo  $k = q_1 \times q_2$ . Os  $w_j$  podem ser inicializados, ou seja, podem-lhes ser dados coordenadas, aleatoriamente ou forçar a pertencerem ao plano definido pelas componentes principais, portanto através da Análise de Componentes Principais (PCA). Basicamente, a inicialização PCA pode ser considerada como um atalho para alocar os nós próximos do local onde os dados estão presentes. Os protótipos podem ser vistos como “botões”, “costurados” no plano de componentes principais num padrão regular e a técnica dos SOM tenta “dobrar” o plano para que os “Botões” se aproximem dos pontos de dados da melhor forma possível. (Hastie, Tibshiran, & Friedman, 2008, p. 529). Todos os nós do mapa SOM estão ligados a outros nós adjacentes e a essa conexão entre os nós é definida por uma relação de vizinhança. Ao contrário do algoritmo *K-means*, onde é atualizado apenas o vetor dos nós vencedores ou seja das melhores unidades correspondentes (BMU), no algoritmo SOM todos os nós adjacentes são atualizados empurrando-os para o treino atual, esta ação é que permite à rede SOM se adaptar ao mapa topológico dos dados originais (Moura, Neves, Reis, & Correia).

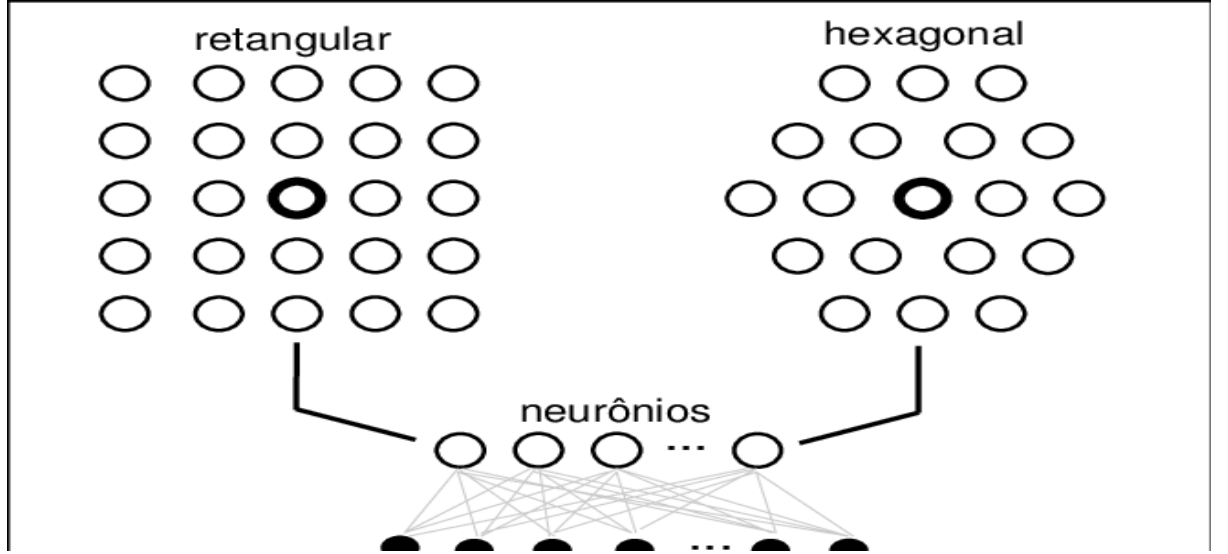


Figura 15: Exemplo de representação de Mapas SOM's (Retangular e Hexagonal).

(Gonçalves, L De, Netto, & Zullo Junior, 2019)

Após a seleção da BMU, o algoritmo original SOM assume que os pesos dos nós adjacentes devem ser movidos para próximo dos dados por via da seguinte atualização:

$$W_j(t+1) = W_j(t) + h_{BMUj}(t) \times (X_i - W_j(t)) \quad (21)$$

onde  $h_{BMUj}$  é a função de vizinhança,  $t$  é o passo computacional, e BMU será o índice do nó  $W_{BMU}$  que será o nó vencedor, ou seja, o nó com a menor distância a  $X_i$ ,  $BMU = \operatorname{argmin}\{d(X_i, W_j)\}$ , onde  $d$  poderá ser qualquer vetor de distância, sendo que a mais usada no algoritmo SOM é a distância euclidiana. A função de vizinhança mais usada é a seguinte:

$$h_{BMU,j}(t) = \alpha(t) e^{-\frac{\|l_{i,j} - l_{BMU}\|^2}{2\sigma^2(t)}} \quad (22)$$

onde  $\alpha$  é uma função escalar decrescente (taxa de aprendizagem),  $l_{i,j}$  e  $l_{BMU}$  são as localizações dos respectivos nós no mapa SOM e  $\sigma$  é outra função decrescente. O algoritmo acima descrito utiliza todas as observações uma de cada vez, sendo para tal um processo que poderá demorar muito tempo na sua implementação. Para tal pode-se implementar uma outra versão, a versão em lote (*Batch*) que expõe todos os nós do SOM a todos os

dados em cada época (passo computacional  $t$ ). Todos  $X_i$  são comparados com todos os nós  $W_j$ , ao mesmo tempo, para que posteriormente sejam selecionadas “cópias dos  $X_i$ ” por nó, nó que foi considerado o BMU desses  $X_i$ . De seguida, estas cópias de  $X_i$  selecionadas formam uma sublista associada ao referido nó e é calculada a média dessas entradas selecionadas sendo associada às coordenadas do BMU que deverão ser atualizadas através de (Kohonen, 2013):

$$W_{BMU}^*(t) = \frac{\sum_{j=1}^N n_j(t) h_{BMU,j}(t) X_{mean,j}(t)}{\sum_{j=1}^N n_j(t) h_{BMU,j}(t)} \quad (23)$$

onde  $X_{mean,j}$  é a média dos dados de entrada que foram selecionados e  $n_j$  o número de dados de entrada selecionados.

Existem ainda outros parâmetros de treino que devemos ter em conta antes de iniciar o treino do algoritmo: o Tamanho do Mapa (número de nós), Forma do Mapa, Função de Vizinhança, Raio de Vizinhança, a Taxa de Aprendizagem e a Duração do Treino. (Versanto, 2000)

Para minimizar o esforço de procura, existem valores padrão, tal como para dimensão do mapa. Caso o utilizador esteja interessado em um valor para o número de nós como ponto de partida, pode aproximar esse valor a  $5\sqrt{n}$  onde  $n$  é a dimensão da amostra (Versanto, 2000). A forma do mapa usualmente é bidimensional retangular ou hexagonal figura 15, e, o raio de vizinhança, tal como a taxa de aprendizagem, é uma função monotonicamente decrescente no tempo, sendo que, o valor do raio de vizinhança inicialmente depende do tamanho do mapa e o seu valor final é 1. Já a taxa de aprendizagem começa com valor inicial de 0.5 e termina em aproximadamente 0 (zero). O comprimento de treino é medido em cada época, uma época corresponde a uma passagem pelos dados e o seu valor depende da relação entre o número de unidades do mapa e o número de amostras. (Versanto, 2000)

A escolha do melhor modelo SOM pode-se basear principalmente no mapa que apresentava os menores erros topológico e de quantização. O erro topológico foi proposto por (Kiviluoto, 1995) com o objetivo de perceber se as propriedades topológicas dos mapas estariam ou não a ser preservadas e é descrito a partir da seguinte fórmula:

$$TE = \frac{1}{n} \sum_{i=1}^n U(X_i) \quad (24)$$

Onde  $u(X_i)$  será igual a 0 caso o primeiro e segundo BMU's forem adjacentes e igual 1 caso não forem adjacentes. (Kiviluoto, 1995)

O erro de quantização permite-nos perceber como é que o SOM está a representar os dados de entrada. Um valor alto para o erro de quantização indicaria que precisamos de mais unidades no mapa (nós), ou, se os “nós” já forem suficientes então precisaríamos de mais etapas de treino (Lobo V. J., 2002). A média do erro de quantização é frequentemente usada para avaliar a representação do SOM sobre os dados de entrada, pois, fornece-nos a média da distância entre as unidades de dados de entrada e o seu representante SOM. (Kiviluoto, 1995)

$$QE = \frac{1}{n} \sum_{i=1}^n \|X_i - W_{BMU}\|^2 \quad (25)$$

Normalmente a fase de treino dos SOM é dividido em duas partes sendo que, na fase inicial do treino é feito um treino áspero onde o raio de vizinhança e a taxa de aprendizagem é maior e no final do treino é feito um treinamento mais fino com os parâmetros acima citados menores (Versanto, 2000). Podemos afirmar que, na primeira fase de treino, a chamada treino áspero os neurónios após serem inicializados com valores aleatórios são “espalhados” e puxados em direção a área geral (no espaço de entrada) onde irão permanecer, esta fase é também chamada de fase de desdobramento e após a sua conclusão a forma geral da rede/grade/mapa no espaço de entrada estará definida e então segue-se para a fase do ajuste fino que tem a finalidade de combinar os neurónios tanto quanto possível aos dados de entrada diminuindo assim o erro de quantização. (Lobo V. J., 2002)

No entanto, para a presente tese foram usados os mapas de *Kohonen* para classificar os novos registos acústicos. Parecendo um contrassenso, usar a o nó na rede SOM como elemento classificador de dados acústicos através da proximidade de um dado novo a um dos nós da rede, pensa-se ser uma solução a considerar dada a especificidade dos dados desta tese.



## 4.3 Técnicas de Previsão

As tarefas da *data mining* podem ser divididas em dois grupos que são: Descrição e Previsão. A descrição responsabiliza-se por identificar regras que caracterizam os dados analisados e a previsão utiliza determinados atributos dos dados para prever o valor de uma outra variável. Por outras palavras, segundo (Berry & Linoff, 2000), a descrição permite aumentar o conhecimento acerca dos dados e a previsão permite suportar o processo de tomada de decisão. (Santos & Ramos, 2017)

### 4.3.1 Classificação

Atualmente, uma das tarefas principais da *data mining* e consequentemente a mesma que está relacionada com o objetivo da Tese é a tarefa de Classificação. A classificação consiste em “examinar” um conjunto de características (*features*) de um objeto recém apresentado e atribuí-lo a uma das classes pré-definidas (Berry & Linoff, 2004), classes essas, que representam o conjunto de valores possíveis explícitos nos dados analisados para o atributo alvo do processo de classificação (o atributo de saída). (Santos & Ramos, 2017)

Os objetos a serem classificados são geralmente representados por registos em tabela na base de dados (Berry & Linoff, 2004), contudo, é importante separá-los em dados de treino, validação e teste. Os dados de treino são usados para treinar o classificador, os dados de validação servem para verificar o desempenho do classificador e os dados de teste serão aqueles que não foram usados no processo de treinamento e servem para efetuar a validação final do classificador. Poderá existir alguma confusão entre os dados de treino e os dados de validação, suas definições e aplicações, contudo, dependem da técnica usada para conceber o classificador, os dados de validação não são usados durante o treinamento, ou seja, os parâmetros do classificador não são derivados deles, mas estão disponíveis para durante o treinamento para verificar o desempenho do classificador. (Lobo V. J., 2002, p. 138)

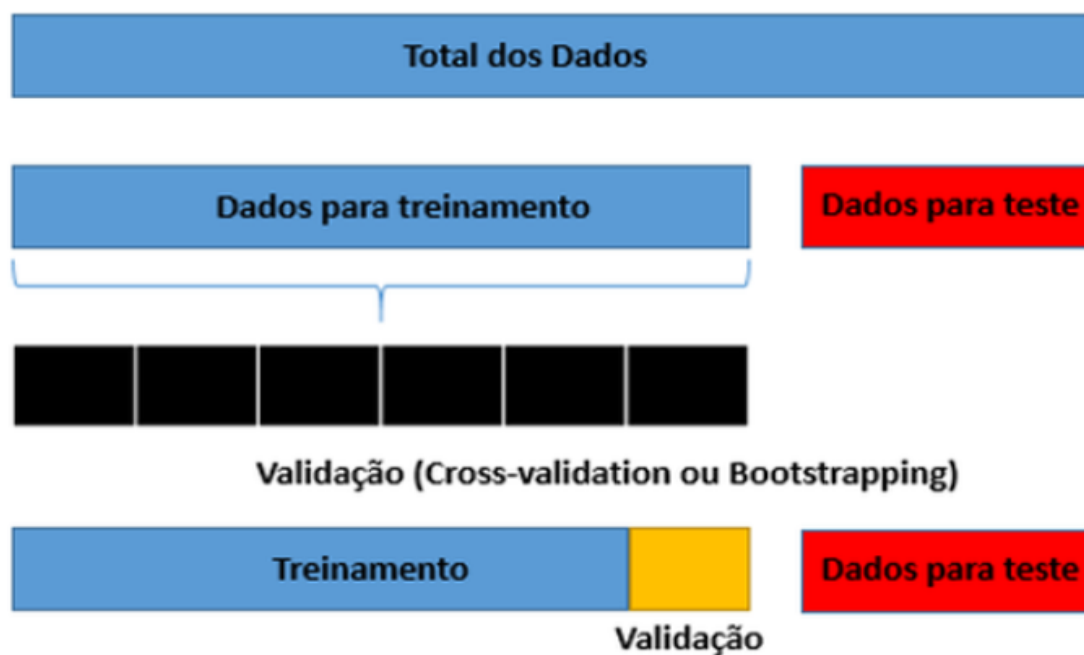


Figura 16: Exemplo de dados de treino, validação e teste.

É importante que se tenha o cuidado com o fenómeno *overfitting* que se traduz no facto do classificador ajustar-se perfeitamente aos dados de treinamento mas ter um péssimo desempenho quando usados com novos dados, isso também deve-se ao facto de se usar demasiados atributos o que deixa o classificador muito detalhado, portanto, reduzir os atributos poderá de algum modo melhorar a generalização do classificador. (Lobo V. J., 2002, p. 44)

A classificação é considerada uma tarefa de aprendizagem supervisionada, o que quer dizer que o atributo e as classes que irão nos conduzir para o processo de classificação são conhecidos à partida e são chamados de conjunto de dados de treino que se baseiam em exemplos pré-classificados que vão treinar o classificador (Berry & Linoff, 2004). A identificação das propriedades comuns entre os registos que permitam associá-los a uma determinada classe, passa por uma exploração do conjunto de valores possíveis contidos nos atributos a analisar que constituem os dados de entrada. (Santos & Ramos, 2017)

A tarefa de classificar é uma tarefa humana imperativa, para podermos nos entender e comunicar com o mundo real nós estamos sempre a classificar, categorizar e classificar outra vez. Alguns exemplos de tarefa de classificação em negócio e pesquisa incluem (Larose, 2005):

- Classificação dos candidatos de crédito como baixo, médio e alto risco;
- Diagnosticar se uma determinada doença está presente;
- Identificar se o testamento foi escrito pelo falecido real ou fraudulentamente por outra pessoa;
- Identificar se determinados comportamentos financeiros ou pessoal indicam ou não uma possível ameaça terrorista.

Existe uma grande diversidade de técnicas de *data mining* que são concretizadas através de diferentes algoritmos, algumas dessas técnicas são: árvores de decisão, regressão linear, redes neuronais e vizinhos mais próximos (k-vizinhos). Na presente tese apenas descreveremos os k-vizinhos e as redes neuronais, estas técnicas estão devidamente descritas em literaturas como de (Larose, 2005), (Berry & Linoff, 2004) ou (Santos & Ramos, 2017).

#### **4.3.2 K-vizinhos**

Ao analisar os resultados e na hora de classificar o novo dado do modelo é necessário usar a técnica k-vizinhos mais próximos (*KNN*) que será importante para saber quem são os neurónios mais próximos do novo dado e classificá-lo consoante o grupo (classe) mais frequente entre os vizinhos mais próximos. Podemos dividir o método em 3 passos simples para classificar um novo objeto:

1. Encontrar os  $k$  vizinhos mais próximos;
2. Saber a que grupo ou classe pertencem;
3. Classificá-lo com a classe mais frequente dos seus vizinhos;

A proximidade entre os objetos do conjunto de treino e um novo objeto é definida por uma métrica que dependem muito do tipo de variáveis a ser usadas (numéricas, categóricas, etc). Normalmente é usada a distância euclidiana, embora tendo a desvantagem de atribuir o mesmo peso para todas as variáveis, por outras palavras, todas as variáveis

têm a mesma importância relativa. A normalização destes dados antes do cálculo é de extrema importância pois permite prevenir que as variáveis com maior variabilidade não sejam consideradas mais importantes do que as variáveis com menor variabilidade (Han & Kamber, 2012).

Quantos vizinhos mais próximos ( $k$ ) analisar? De facto, para a escolha do  $k$  não existe uma solução óbvia pois escolhendo um  $k$  muito pequeno é possível que a classificação possa ser afetada por *outliers* ou encontrar muitos ruídos (observações incomuns) na vizinhança do objeto a classificar (Larose, 2005, p. 106), da mesma forma que, caso o  $k$  for um valor muito grande existe a possibilidade de serem incluídos objetos que fazem parte do conjunto de treino que não estão na zona de interesse para a classificação. Portanto, o valor de  $k$  depende do problema em questão e para uma escolha mais adequada recorre-se a elaboração de um gráfico onde estão representados vários  $k$  e a probabilidade de má classificação dos mesmos (Han & Kamber, 2012).

Além do método KNN ser ter demonstrado muito poderoso ao longo dos anos, acarretam algumas desvantagens por ser computacionalmente intensivo porque necessita calcular as distâncias todas as vezes que o programa é corrido e quando existem dados em dimensões elevadas o vizinho mais “próximo” pode estar muito longe, este fenómeno chama-se maldição da dimensionalidade. Por outro lado, o KNN tem a vantagem de ser de fácil perceção e implementação pois possui um algoritmo simples como iremos descrever a seguir:

1. Seja  $K$  o número de vizinhos mais próximos selecionado;
2. Para cada objeto do conjunto de teste  $X_0$ :
  - a. Computar a distância  $d_{(X_0, X_i)}$  onde  $(X_i^t, Y_i) \in D$ ;
  - b. Selecionar  $D_{X_0} \subseteq D$ , o conjunto dos  $K$  objetos de treino mais próximos de  $X_0$ ;
3.  $Y_0 = \operatorname{argmax}_K \sum_{(X_0^t, Y_i) \in D_{X_0}} I(Y_i, C_k)$  onde:
  - a.  $I(Y_i, C_k) = \begin{cases} 1 & \text{se } Y_i = C_k \\ 0 & \text{c. c} \end{cases}$
  - b.  $\sum_{(X_0^t, Y_i) \in D_{X_0}} I(Y_i, C_k) = \eta_K$  conta o número de objetos em  $D_{X_0}$  com classe  $C_k$ .

### 4.3.3 Redes Neurais Artificiais

O cérebro humano possibilita que as pessoas aprendam a partir da experiência, por outro lado, os computadores digitais geralmente se destacam em seguir instruções explícitas repetidas vezes. As redes neurais artificiais aparecem com o intuito de superar essa limitação modelando em um computador digital, as conexões neurais do cérebro humano.

As redes neurais são uma classe de ferramentas poderosas muito aplicadas em diversas áreas para solucionar problemas de previsão, classificação e agrupamento. Desde previsão de series temporais no mundo financeiro até ao diagnóstico de condições médicas, desde a identificação de *cluster* de clientes valiosos até a identificação de transações fraudulentas com cartões de crédito (Berry & Linoff, 2004). São compostas por um conjunto de unidades organizadas em níveis, as unidades (nós) encontram-se conectadas através de ligações onde cada uma das ligações tem associada um peso ajustável. Podemos dividir as diversas unidades que constituem a rede em três grandes grupos: Unidades/camada de entrada, as encarregadas de receber os dados a analisar, unidade intermédia ou camada oculta que é um número ilimitado de níveis intermédios onde é feito a maior parte do processamento e a unidade ou camada de saída que transmitem os sinais de saída da rede, transmitindo os possíveis resultados do modelo. (Santos & Ramos, 2017)

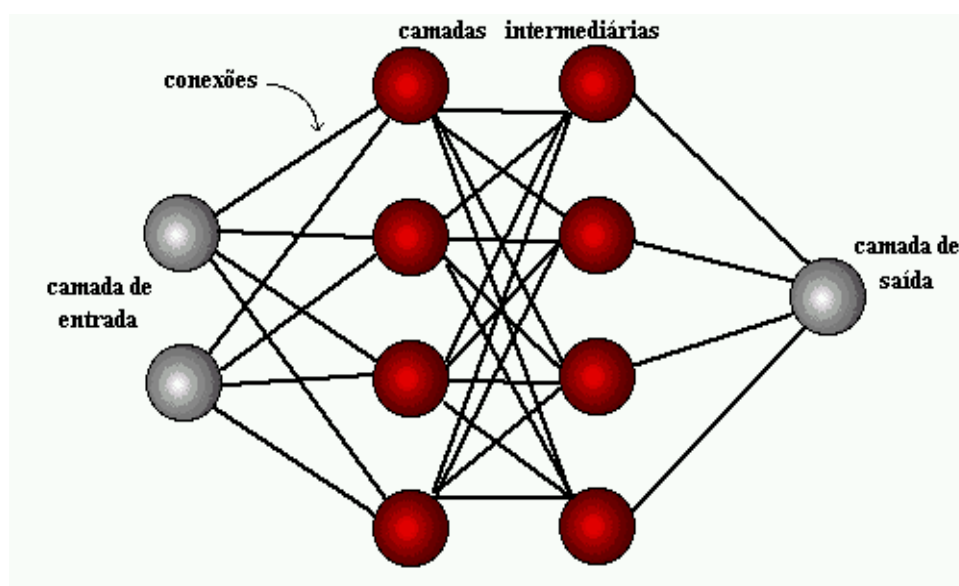


Figura 17: Exemplo de uma Rede Neuronal Artificial organizada por camadas.

Para compreender melhor a lógica do funcionamento das redes neuronais artificiais será necessário introduzir alguns conceitos básicos referentes ao funcionamento do cérebro humano e seus componentes. Os neurónios e suas conexões entre células são de grande importância bem como algumas considerações sobre como se concebe teoricamente o funcionamento matemático. (Haykin, 2001)

O neurónio biológico é a unidade básica do cérebro sendo uma célula especializada na transmissão de informações e está constituído por três partes principais: a soma ou corpo molecular, onde emanam algumas ramificações denominadas de dendritos, e uma outra ramificação descendente da soma, mais extensa, chamada de axónio. Nas extremidades dos axónios encontram-se os nervos terminais. (Haykin, 2001)

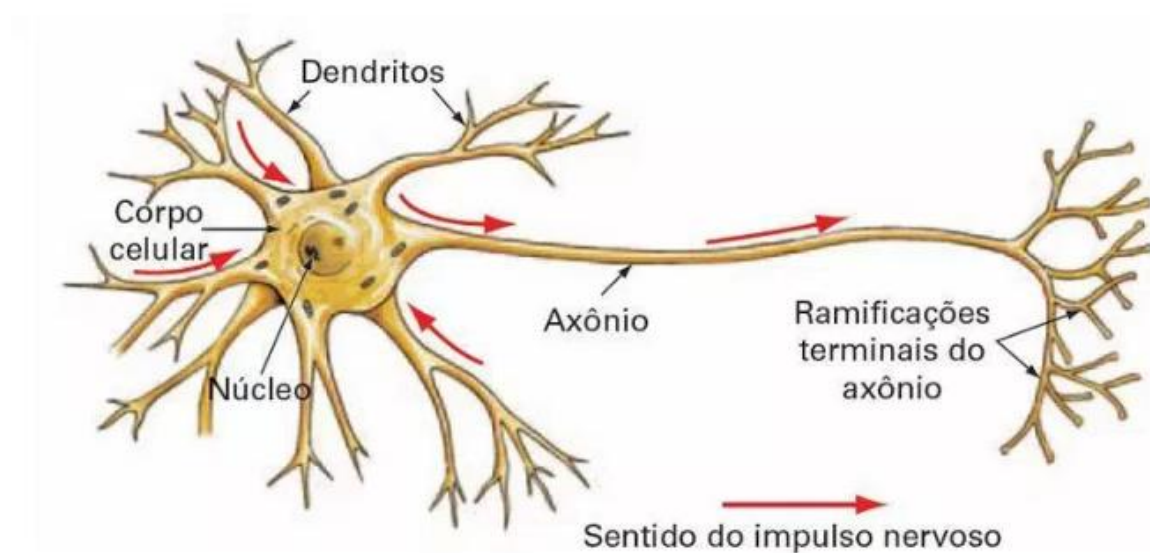


Figura 18: Representação Simplificada do Neurônio Biológico. (Haykin, 2001)

O cérebro humano está constituído por bilhões de neurónios conectados entre eles, o que forma uma enorme rede de comunicação, a rede neural. Os dendritos recebem sinais elétricos provenientes de outros neurónios através das sinapses que são nada mais do que o ponto de contacto entre a terminação axónima de um neurónio e o dendrito de outro. A informação é processada no corpo celular e geram-se novos impulsos passando pelos axónios e atingindo os dendritos dos neurónios seguintes. Notar que, para cada neurónio

que recebe o sinal pode esse sinal ser transmitido, amplificado ou atenuado, pois existe um certo valor limite (*Threshold*) que o sinal é bloqueado caso seu valor seja inferior ao *Threshold*. Cada neurónio está associado a um peso pelo qual o sinal é multiplicado. (Haykin, 2001)

Inspirados no neurónio biológico, pesquisadores desenvolveram um modelo de neurónio matemático que se tornou a base da inteligência artificial. (Haykin, 2001)

O modelo matemático do neurónio biológico é composto por: sinais de entrada, que são multiplicados pelo seu fator excitatório que são os pesos sinápticos, e posteriormente uma função de ativação, que definirá com base nas entradas e pesos, qual será a saída do neurónio. (Haykin, 2001)

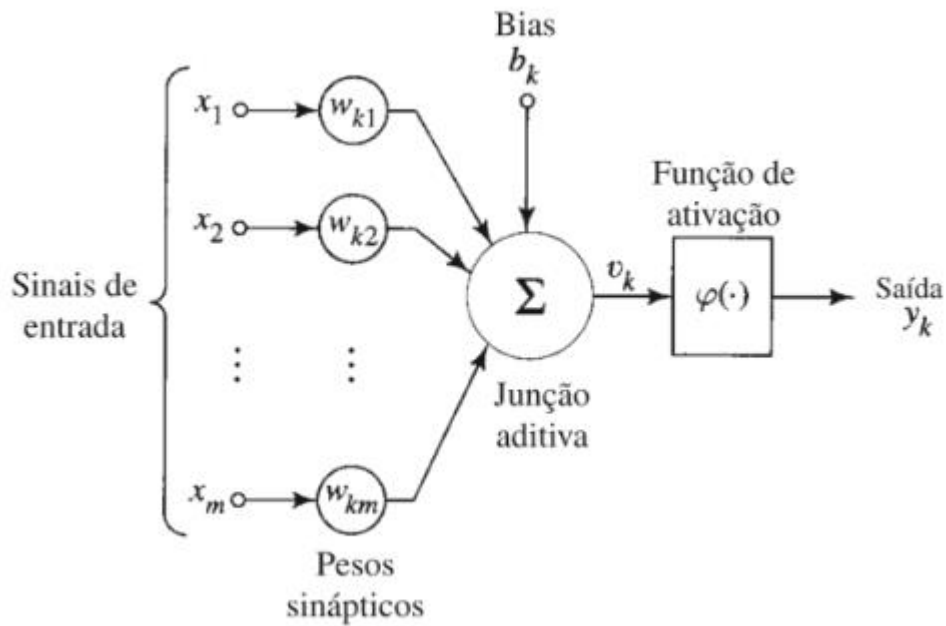


Figura 19: Modelo não linear de um neurónio. (Haykin, 2001, p. 36)

Em termos matemáticos podemos descrever um neurónio K com o seguinte par de equações:

$$U_k = \sum_{j=1}^m W_{kj} X_j \quad (26)$$

$$Y_k = \varphi(U_k + b_k) \quad (27)$$

Onde  $X_1, X_2, \dots, X_m$  são os sinais de entrada;  $W_1, W_2, \dots, W_{km}$  são os pesos sinápticos de um neurónio K;  $U_k$  é a saída do combinador linear devido aos sinais de entrada;  $b_k$  é o bias;  $\varphi(.)$  é a função de ativação e  $Y_k$  é o sinal de saída do neurónio. (Haykin, 2001, p. 38)

O *bias*  $b_k$  é usado para aplicar uma transformação em  $U_k \rightarrow V_k = U_k + b_k$ , sendo o *bias* positivo ou negativo, a relação entre o campo local induzido  $V_k$  do neurónio K e a saída do combinador linear  $U_k$  terá a seguinte forma tal como apresenta a Figura 20.

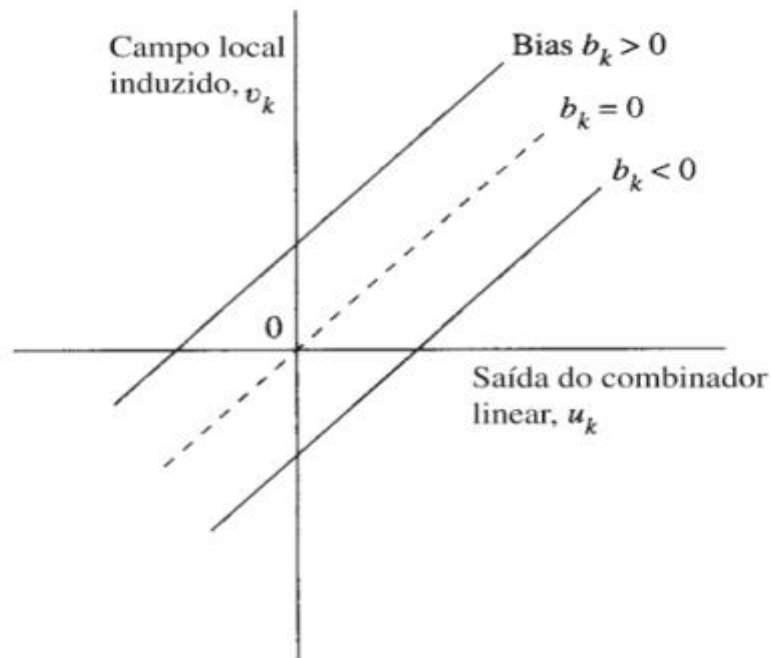


Figura 20: Transformação afim produzida pela presença de um bias. (Haykin, 2001, p. 37)

Como podemos observar cada elemento do neurónio artificial é importante para a qualidade dos resultados na camada de saída. Passando agora para as funções de ativação, podemos afirmar que as funções de ativação têm um papel muito importante nas redes neuronais por permitirem que pequenas mudanças nos pesos e *bias* causem apenas pequenas alterações no seu *output* (saída) e não na restante rede, as funções de ativação



decidem se o neurónio deve ser ativado ou não, ou seja, basicamente elas medem se a informação recebida é relevante para ser fornecida ou ignorada. A função de ativação é uma transformação não linear feita ao longo do sinal de entrada o que permite que saída transformada seja enviada para a próxima camada como entrada, quando não existe função de ativação os pesos e os bias fazem uma transformação linear o que limita a rede na execução de tarefas mais complexas como classificação de imagens, classificação de sons, entre outros, por este motivo o uso de funções de ativação é de extrema importância. (Haykin, 2001)

Na presente dissertação usou-se para todas camadas ocultas a função de ativação de tipo ReLu e para a camada de saída, por se tratar de um problema de classificação usou-se a função de ativação *softmax*.

A função de ativação ReLu está definida matematicamente como:

$$f(x) = \max(0, x) \quad (28)$$

Atualmente, a função ReLu é uma das mais populares para projeção de redes neurais, possui a grande vantagem de não ativar todos neurónios ao mesmo tempo e caso a entrada seja negativa, ela será convertida a zero, além de ser computacionalmente mais leve por possuir operações matemáticas mais simples, a sua grande desvantagens é a sua linha horizontal (para X negativo) pois o gradiente pode ser zero para essa regiões ReLu o que significa que os pesos não serão ajustados durante a descida que pode fazer com que vários neurónios da rede “morram” não sendo mais ativados e tornando uma parte substancial da rede passiva (Sharma V, 2017). Por outro lado, a função de ativação *Softmax* usada apenas nas camadas de saídas é muito usada em redes de classificação pois força a saída de uma rede neuronal a representar a probabilidade dos dados serem de uma das classes definidas.

$$Y_i = \frac{e^{Z_j}}{\sum_{j \in grupo} e^{Z_j}} \quad (29)$$



## 5 Aplicação

Numa tentativa de avaliar os SOM's como classificadores de navios através da sua assinatura acústica, iremos comparar a precisão desta técnica com a classificação através de RNA's. Relembra-se que os mapas SOM's têm como objetivo principal agrupar os dados naturalmente e que dessa forma se pode também usar como classificador se tiver um número de neurónios adequado para tal.

A base de dados acústicos usados para elaboração deste trabalho foi fornecida pelo autor da tese "*Ship Noise Classification a contribution to prototype based classifier design*" que vem colaborando com a Esquadilha de Submarinos da Marinha Portuguesa em reconhecimento de ruído de navios desde 1990. Várias gravações foram feitas ao longo do tempo e guardadas tanto para fins de investigação bem como para fins militares, razão pela qual a informação descritiva (tipos de navios e ambientes na qual foram feitas as gravações) da base de dados usada na presente tese não poderão ser discriminados.

Esta base de dados acústicos onde as técnicas foram aplicadas é constituída por 73 ficheiros áudio com aproximadamente um minuto de áudio, correspondendo a 33 (numerados de 1 a 33) navios diferentes, registados em 11 ocasiões/situações diferentes (catalogados alfabeticamente de *a* a *k*). Embora as gravações tenham sido feitas por gravadores de alta qualidade, estas ocasiões irão influenciar a sua classificação devido ao ruído inerente a cada um dos locais de gravação, sendo, portanto, importante ter isso em conta aquando da classificação, visto que o objetivo principal é identificar um certo navio num local diferente dos locais dos registos históricos. Não possuindo um número elevado de registos e dado que o som produzido pelos motores de cada navio são periodicamente idênticos, decidiu-se segmentar os ficheiros em 4 partes com a duração aproximada de 16 segundos cada, para que se possua mais elementos com assinaturas acústicas de cada navio, permitindo uma melhor aplicabilidade de técnicas assentes em redes neuronais.

A todos estes ficheiros foram extraídas 193 componentes/características através da extração de características descritas no Capítulo 3. Para tal, recorreu-se a um pacote

de algoritmos, habitualmente usados por investigadores aquando da utilização do software Python para classificação de dados através de registos acústicos, o *librosa*. (Mcfee, 2015)

Posteriormente, efetuou-se uma divisão manual dos dados em três grupos: treino, validação e teste. Para o grupo de teste, foram separados todos os ficheiros/segmentos que provinham do ambiente catalogado com a letra *a*, contendo todos os navios, no entanto, apenas aqueles que continham pelo menos dois registos em ambientes diferentes, perfazendo um total de 60 registos (15 ficheiros  $\times$  4 segmentos). Para o grupo de validação, que posteriormente se adicionará ao grupo de treino, foram isolados os ficheiros de todos os navios registados no ambiente *c*, sendo selecionados apenas os que continham registos em pelo menos três ambientes diferentes, perfazendo um total de 28 registos (7 ficheiros  $\times$  4 segmentos). Os restantes 204 elementos (51 ficheiros  $\times$  4 segmentos) serão agrupados no conjunto de treino.

Em primeira mão, vários SOM's foram treinados usando o grupo de treino, usando um pacote bastante completo de treino e visualização de SOM's, o SOMPY<sup>11</sup>. De notar que este pacote de algoritmos usa o treinamento em lote (*batch*). Dentre todas as possibilidades, começou-se por considerar a inicialização através da utilização de neurónios aleatoriamente gerados que estivessem contidos no hiper-plano definido pelas duas primeiras componentes principais. Esta configuração permitirá um rápido treinamento e uma razoável qualidade na divisão dos dados, que será útil para uma primeira definição de quantos neurónios a utilizar na rede SOM. Usando os dados originais (193 variáveis) não normalizados, normalizados por Min-Max ou normalizados por Normalização Standard, foram treinados vários mapas planares, com as dimensões pertencentes aos seguintes domínios inteiros:  $[18,21] \times [18,21]$ ;  $[9,10] \times [9,10]$  e  $7 \times 10$  (medida esta, recomendada automaticamente por uma função do SOMPY baseando-se na medida referenciada no subcapítulo 5.2.2). Todos os resultados destas primeiras simulações poderão ser consultados nos Anexos A e B, para grelhas hexagonais e retangulares, respetivamente. A qualidade do mapa, neste caso, irá refletir-se nos valores da precisão obtida nos dados

---

<sup>11</sup> SOMPY – Pacote criado e mantido principalmente por: Vahid Moosavi @sevamoo; Sebastian Packmann @sebastiandev; Iván Vallés @ivallesp. Podendo ser consultado em <https://github.com/sevamoo/SOMPY>

de validação. Esta classificação é efetuada localizando o BMU do SOM para cada dado de validação e a classe atribuída a esse dado corresponde à classe mais frequente relativamente a esse BMU. No caso de empate de classes, decidiu-se, nesta aplicação, que em vez de atribuir aleatoriamente uma das classes se atribuiria a classe correspondente ao ficheiro registado em primeiro lugar. Estas primeiras simulações permitiram apurar qual a dimensão do mapa a utilizar que possivelmente permitirão obter melhores resultados, dimensão  $7 \times 10$ , e qual a grelha, a hexagonal. A próxima fase, consiste em fixar as duas variáveis descritas anteriormente e alternar o número de iterações do treinamento em bruto e do treinamento fino juntamente com o tipo de normalização. Para cada uma dessas variantes, treinou-se 10 mapas SOM diferentes, porém, com inicialização aleatória das coordenadas dos neurónios. A precisão para os dados de validação melhor foi registada para um mapa SOM com um treinamento em bruto de 20 iterações e treino fino de 50 iterações, quando os dados são normalizados por Min-Max. Com estas configurações definidas e fixadas, voltou-se a treinar, mas desta vez, com os dados de treino e os de validação conjuntos e apurou-se a precisão da classificação nos dados de teste. Na Tabela 1, apresentam-se os resultados obtidos para 10 mapas treinados, contendo a precisão obtida tanto nos dados de treino como nos dados de teste.

Modelo n.º	Precisão de treino	Precisão de teste	Erro topográfico	Erro de quantização
1	60,3%	50%	0,15	0,06
2	65,5%	51,7%	0,19	0,06
3	58,2%	36,7%	0,22	0,07
4	58,2%	51,7%	0,32	0,06
5	62,1%	31,7%	0,38	0,06
6	60,7%	51,7%	0,20	0,06
7	58,2%	58,3%	0,14	0,06
8	63,4%	58,3%	0,28	0,06
9	62,1%	41,7%	0,22	0,06
10	58,2%	58,3%	0,11	0,06

Tabela 1: Resultados de 10 mapas treinados, com dimensão  $7 \times 10$ , grelha hexagonal, 20 iterações de treino bruto e 50 iterações de treino fino.

Observando essa mesma tabela, será importante notar que 7 modelos em 10 obtiveram valores de precisão de teste superiores a 50%. Na figura 21, pode-se observar um mapa

de componentes do modelo 8, cuja precisão de teste foi a melhor, e onde se pode visualizar em cada neurónio a classe que o distingue, ou seja, a classe que seria atribuída a cada novo dado, caso o seu BMU fosse o neurónio identificado. O valor  $-50$  foi apenas inserido para que os nós cuja classificação corresponderia a um áudio desconhecido no mapa de componentes tivesse uma cor associada que fosse escura e não se destacasse.

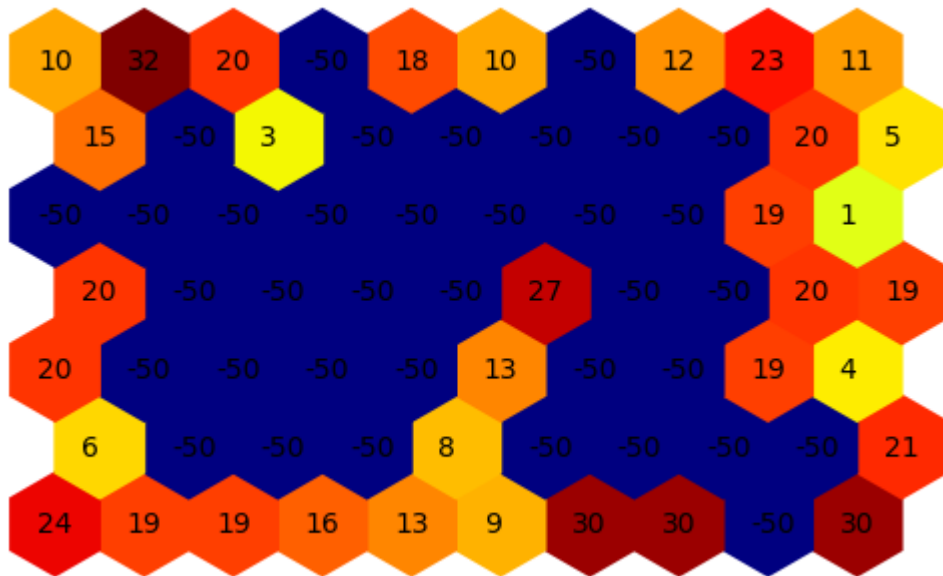


Figura 21: Mapa de componentes contendo a classificação predominante como anotação.

Para se ter a noção de como a classificação foi feita para este último modelo, no Anexo B pode-se observar as possíveis classes por neurónio, juntamente com o número de dados dessa classe ‘presentes’ nesse neurónio bem como a percentagem de segmentos do total relativamente a essa classe estão ‘inseridos’ nesse neurónio, que iremos designar por confiança de determinada classe. No Anexo C, podem-se ver os restantes mapas de componentes bem como o mapa de componentes do SOM cuja configuração é idêntica excluindo o facto de a inicialização ser feita através de PCA.

Em segundo lugar, procedeu-se à escolha de uma rede neuronal de uma camada interna/escondida. Cada RNA contém uma camada de entrada contendo 193 neurónios de entrada, correspondendo ao número de variáveis, uma camada escondida com valores pertencentes ao conjunto de números inteiros  $\{4, 8, 16, 32, 64, 128\}$  e uma camada de

saída com 34 neurónios correspondendo aos 33 navios e uma mais para a classe ‘navio desconhecido’. Todas as funções de ativação foram as funções *ReLU*, exceto as de saída que foram as *Softmax* por se tratar de uma classificação. O número de iterações que foram consideradas foram 50, 100 e 200, não sendo necessário aumentar esse número, pois foi observado que o treino das RNA entrava no fenómeno de sobre-aprendizagem para valores superiores. Contemplando os diferentes tipos de normalização, Min-Max, Standard e sem normalização, apurou-se que o número de neurónios internos/escondidos mais indicado seriam os 64 neurónios. Tornou-se a simular 10 vezes para cada número de iterações indicado anteriormente, contemplando novamente as diferentes normalizações e concluiu-se que as 100 iterações correspondem a um número cuja precisão de validação era boa e estável sem que houvesse lugar a sobre-aprendizagem.

Na figura 22, pode-se visualizar um dos treinos efetuados para 200 iterações que demonstram uma certa sobre-aprendizagem quando comparados os erros de validação com os de treino, ou seja, um aumento do erro de validação, podendo para tal afetar a precisão de validação.

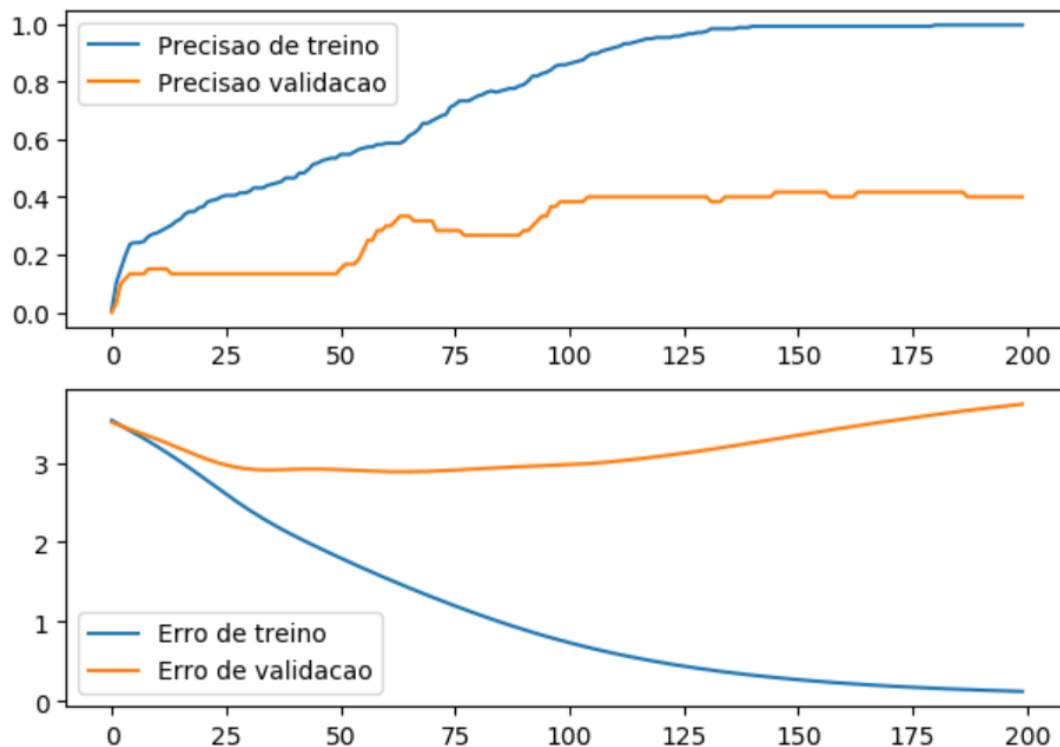


Figura 22: Precisão de treino e Precisão validação (cima) Erro de Treino e Erro de

### Validação (baixo).

No entanto, para realmente compararmos as capacidades do classificador RNA com o classificador SOM, procedeu-se para um total de 64 neurónios da camada interna, vários treinos de RNA's com os dados de treino conjuntamente com os dados de validação e foram testados com os dados de teste. No Anexo D, podem-se consultar todos os resultados obtidos nessas simulações. De entre todos os resultados obtidos para os diferentes modelos RNA, destaca-se o facto de a precisão de teste mais elevada registada foi a de 48,3 %, obtendo, portanto, apenas valores de precisão inferiores a 50%. Para que se possa fazer uma análise mais aprofundada desse resultado, na Tabela 2 apresentam-se todos as classificações cuja saída (probabilidade de ser de determinada classe) da classe atribuída foi inferior a 50%. Nessa tabela apresentam-se apenas as probabilidades associadas às classes que seriam escolha acertada se recorrêssemos à segunda classe mais provável.

Classificação Real	Classificação Prevista – Probabilidade obtida no neurónio correspondente	2.ª Classificação Prevista – Probabilidade obtida no neurónio correspondente
4	31 – 36,2%	4 – 18%
4	31 – 48,5%	4 – 16,7%
4	31	17
18	5	19
22	30	19
22	30	19
22	30	19
22	30	13
27	13 – 49,7%	27 – 34,7%
27	13 – 42,7	27 – 38,9%
32	25	30
32	2	9
32	13	2
32	2	9

Tabela 2: Comparação entre a Classificação Prevista, a Classificação Real e a 2.ª Classificação mais provável dos dados mal classificados.

Nessa mesma tabela, pode-se verificar que apenas em 4 momentos a segunda opção seria a classificação correta e que nas restantes tanto a primeira como a segunda opção seriam mal classificadas.



## 6 Conclusões

No presente trabalho pretendeu-se comparar duas técnicas de classificação de dados acústicos, Redes Neurais Artificiais e Mapas Auto-Organizados, mais propriamente na identificação de navios através da sua assinatura acústica por parte de um marinista.

A primeira técnica tem como finalidade esse mesmo objetivo, o de efetuar uma previsão, sendo uma das mais utilizadas na atualidade para a classificação de sons. A segunda, é uma técnica que foi criada para outro fim, o de agrupar dados pelas suas características, sendo, portanto, uma técnica de agrupamento. Tendo em conta as características inerentes aos SOM's, ou seja, o facto de este agrupamento ser efetuado através da utilização de uma grelha contendo vários nós interligados que se ajustam aos dados, podem-se adaptá-los e convertê-los em classificadores. Para podermos classificar com esta técnica, temos de atribuir uma classe a cada um dos neurónios (ou ausência de classe) e perante um dado novo é necessário calcular o nó da grelha mais próximo (através da técnica de k-vizinhos) deste e, desse modo, classificar de acordo com a classe atribuída ao nó. As duas técnicas são consideradas redes neuronais, dessa forma, podem padecer de um mesmo problema, ausência de dados suficientes, tendo sido para tal encontrado uma solução através da segmentação de cada registo acústico em 4 partes, que conterão individualmente teoricamente a mesma informação da totalidade do registo, tendo em conta que é um som produzidos por um navio (motor, hélices, etc...)

Para que se possa implementar cada uma das técnicas (SOM's e RNA's) e apurar a qualidade destas, é necessário em primeiro lugar extrair as componentes de cada áudio através de técnicas baseadas nas transformadas de Fourier. Estas permitem que de um só registo contínuo do som de um navio se extraiam 193 componentes diferentes.

Analisando os resultados obtidos no capítulo anterior, apurou-se que para os dados de treino o melhor classificador é de facto aquele destinado a classificar, as RNA's. No entanto e inesperadamente, na maioria dos casos (quase na totalidade) a precisão quando se aplica aos dados de teste a adaptação dos SOM's (técnica cuja finalidade seria a de agrupar dados) ao processo de classificação obtiveram melhores resultados do que a

aplicação das RNA's. Estes dados de teste dizem respeito a uma experiência que corresponderia a uma situação mais real (finalidade é identificar navios em qualquer ambiente), ou seja, a tentativa de classificar um dado acústico numa região diferente das regiões onde os registos históricos foram recolhidos. A explicação para este fenómeno poderá residir no facto que as RNA's são treinadas para identificar não só o som emitido pelo navio mas também o ruído envolvente, enquanto os SOM's, como têm a finalidade de separar conjuntos de dados pelas suas diferenças e agregar pelas suas similitudes, de certo modo tem a capacidade de 'ignorar' a presença do ruído ambiente de cada região, permitindo uma melhor classificação.

Será importante notar também que, na transformação dos SOM's em classificadores, a minimização do erro topográfico não irá corresponder a melhores resultados de precisão, ao contrário do que é suposto quando se usa esta técnica para agrupar dados. Outro ponto importante de referir será o de que em ambas as técnicas utilizadas para classificar, a normalização a adotar é relevante, mostrando-se mais eficazes os classificadores aplicados aos dados não normalizados ou a dados normalizados por Min-Max do que quando aplicados a dados normalizados pela normalização Standard, havendo destaque para a Min-Max que apresentou os resultados melhores e mais consistentes.

Numa perspetiva de melhoria do sistema de apoio à decisão, será importante ter em conta o seguinte exemplo: Alocado ao neurónio n.º 69 (vide Anexo B), existem duas classes 'presentes': a Classe 21 e a Classe 30. Um algoritmo direto iria automaticamente escolher a classe 30 para caracterizar um dado acústico novo cujo BMU fosse o nó 69, no entanto, o valor de confiança dessas classes, nesse neurónio, poderão indicar que essa atribuição poderá não ser a mais correta. Tendo em conta que esse nó é BMU de:

- Classe 21 – 3 dados históricos (confiança = 75% dos dados históricos da Classe 21)
- Classe 30 – 4 dados históricos (confiança = 10% dos dados históricos da Classe 30)

apesar de a Classe 30 ter maior 'presença' neste neurónio, a confiança atribuída é de apenas 10%, ou seja, apenas 4 dos 40 dados históricos foram alocados a este neurónio do SOM. Comparando com a confiança da Classe 21 (3 dos 4 dados históricos), pondera-se

se não seria mais correto atribuir esta Classe ao novo dado. Observando o Anexo B, apura-se que estes tipos de situações se repetem noutros neurónios. Portanto, o autor sugere que em vez de se criar um algoritmo que forneça ao utilizador apenas uma classificação para um certo dado novo, seja criado uma alternativa que forneça este tipo de informação a quem estiver perante a responsabilidade de classificar um certo dado acústico e, posteriormente, mediante a sua capacidade e experiência decidir qual das Classes se deve atribuir ao novo registo. Dessa forma, este processo de informação mais detalhada seria um sistema melhorado que apoiaria a decisão de quem estaria a fazer a audição dos registos em tempo real.

Outro aspeto a ter conta seria o da aplicação de filtros adequados aos dados acústicos para minimizar o ruído presente em cada registo e aplicar novamente as duas técnicas referidas para se poder voltar a compará-las, permanecendo ainda a dúvida se uma técnica de agrupamento adaptada (SOM), neste contexto, será na verdade mais eficaz que uma das mais usadas técnicas de previsão (RNA), para identificar um novo registo acústico de um navio.



# Bibliografia

- Alavi, M. e. (2001). *Review: Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues*. MIS Quarterly .
- Araujo, J. F. (2019, agosto 28). [http://www.dsc.ufcg.edu.br/~pet/ciclo\\_seminarios/tecnicos/2012/CicloDeSeminarios2012.1\\_Joseana.pdf](http://www.dsc.ufcg.edu.br/~pet/ciclo_seminarios/tecnicos/2012/CicloDeSeminarios2012.1_Joseana.pdf). Retrieved from [http://www.dsc.ufcg.edu.br/~pet/ciclo\\_seminarios/tecnicos/2012/CicloDeSeminarios2012.1\\_Joseana.pdf](http://www.dsc.ufcg.edu.br/~pet/ciclo_seminarios/tecnicos/2012/CicloDeSeminarios2012.1_Joseana.pdf): [http://www.dsc.ufcg.edu.br/~pet/ciclo\\_seminarios/tecnicos/2012/CicloDeSeminarios2012.1\\_Joseana.pdf](http://www.dsc.ufcg.edu.br/~pet/ciclo_seminarios/tecnicos/2012/CicloDeSeminarios2012.1_Joseana.pdf)
- Berry, M. J., & Linoff, G. (2000). *Mastering Data Mining: The Art and Science of Costumer Relationship Management*. New York: John Wiley and Sons, inc.
- Berry, M. J., & Linoff, G. S. (2004). *Data Mining Techniques: For Marketing, Sales and Costumer relationship Management second edition*. Indianapolis, Indiana: Wiley Publishing, Inc.
- Bridle, J. S., & Brown, M. D. (1974). *Tech. Rep., Joint Speech Research Unit "An experimental automatic word recognition system"*. Ruislip, England.
- Camargo, A. &. (2016). Mineração de dados eleitorais: descoberta de padrões de candidatos a vereador na região da campanha do Rio Grande do Sul. *Revista Brasileira de Computação Aplicada*. 8. 64-73., 64-73.
- Carneiro, F. G. (2019, 06 23). *Percepção do Pitch fantasma utilizando a sirene de Seebeck*. Retrieved from Unicamp: [https://www.ifi.unicamp.br/~lunazzi/F530\\_F590\\_F690\\_F809\\_F895/F809/F809\\_sem1\\_2003/002966Fabio-Knobel-RF08\\_5.pdf](https://www.ifi.unicamp.br/~lunazzi/F530_F590_F690_F809_F895/F809/F809_sem1_2003/002966Fabio-Knobel-RF08_5.pdf)
- Carvalho, H. M. (2014). *Aprendizado de Máquina voltado para Árvores de Decisão. Aprendizado de Máquina voltado para Árvores de Decisão*.

- CHIRP Sonar Technology*. (2019, agosto 28). Retrieved from CHIRP Sonar Technology:  
<https://explore.garmin.com/en-US/chirpsonar/>
- Clodoaldo A. M. Lima, S. M. (2015, agosto 13). *EACH - USP*. Retrieved from Programa de Pos-Graduacao em Sistemas de Informação:  
<http://each.uspnet.usp.br/sarajane/wp-content/uploads/2015/08/aula02.pdf>
- Davis, S. B., & Mermelstein, P. (1980). *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*. IEEE Transactions on Acoustics, Speech.
- Deus, J. D., Pimenta, M., Noronha, A., Peña, T., & Brogueira, P. (2014). *Introdução à Física*. Lisboa: Escolar Editora .
- Duarte, C., & Pereira, A. (1990). *Probabilidades e Estatística*. Setúbal: Instituto Politécnico de Setúbal.
- Goldschmidt, R., & Passos, E. (2005). *Data Mining um Guia Prático*. Editora Campus.
- Gonçalves, M., L De, M., Netto, A., & Zullo Junior, J. (2019, 08 05). Um Sistema Neural Modular para Classificação de Imagens Utilizando Mapas de Kohonen. *Um Sistema Neural Modular para Classificação de Imagens Utilizando Mapas de Kohonen*.
- Han, J., & Kamber, M. (2012). *Data Mining Concepts and Techniques third edition*. New York: Elsevier.
- Hastie, T., Tibshiran, R., & Friedman, J. (2008). *The Elements of Statistical Learning Data Mining, Inference, and Prediction Second Edition*. Stanford, California: Springer.
- Haykin, S. (2001). *Redes Neurais, Princípios e Práticas*. Porto Alegre: Bookman.
- Instituto de Pesquisas da Marinha Marinha do Brasil*. (2019, agosto 28). Retrieved from  
<https://www.marinha.mil.br/ipqm/node/47>

- Kiviluoto, K. (1995). *Topology Preservation in Self-Organizing Maps*. Helsinki University of Technology.
- Kohonen, T. (1982). Clustering, Taxonomy, and Topological Maps of Patterns. *Proceedings of the 6th International conf. pattern Recognition IEE Computer society press* , pp. 114-128.
- Kohonen, T. (1998). The self-organizing map. Neurocomputing. *The self-organizing map. Neurocomputing*, pp. 1-6.
- Kohonen, T. (2013). Essentials of the self-organizing map. Neural networks. *Essentials of the self-organizing map. Neural networks*, pp. 52-65.
- Larose, D. T. (2005). *Discovering Knowledge in Data an Introduction to Data Mining*. New Jersey: wiley-Interscience a John Wiley & Sons inc., Publication .
- Lobo, V. (2019, 06 26). *Nova IMS*. Retrieved from novaims.unl.pt: [https://www.novaims.unl.pt/docentes/vlobo/escola\\_naval/SAD/SAD\\_EN\\_4\\_visualizacao.pdf](https://www.novaims.unl.pt/docentes/vlobo/escola_naval/SAD/SAD_EN_4_visualizacao.pdf)
- Lobo, V. J. (2002). Ship Noise Classification. A contribution to prototype based classifier design.
- Majeed, S. A., Husain, H., & Samad, S. A. (2015). Mel Frequency Ceptral Coefficients (MFCC) Feature Extration Enhancement in the Application of Speech Recognition. *Journal of Theoretical and Applied Information Technology* (pp. 38-56). Malaysia: JATIT & LLS.
- Manzolli, J. (2019, 06 23). *Repositório da Produção Científica e Intelectual da Unicamp*. Retrieved from [Repositório.unicamp.br: http://repositorio.unicamp.br/bitstream/REPOSIP/307395/1/Manzolli\\_Jonatas\\_M.pdf](http://repositorio.unicamp.br/bitstream/REPOSIP/307395/1/Manzolli_Jonatas_M.pdf)
- Mcfee, B. C. (2015). Librosa: Audio and music signal analysis in python . *Proceedings of the 14th Python in Science conference*, pp. 18-25.

- Moura, J. (2009, novembro). *IEEE Signal Processing Magazine*, vol. 26, no. 6. Retrieved from What is signal processing? [President's Message] *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 6-6,: : <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5230869&isnumber=5230792>
- Moura, N. N. (2013, Abril). Detecção e Classificação de Sinais de Sonar Passivo usando Método de Separação Cega de Fontes . *Detecção e Classificação de Sinais de Sonar Passivo usando Método de Separação Cega de Fontes*.
- Moura, R., Neves, J., Reis, F., & Correia, A. (n.d.). Daily Load Profile of Portuguese Electricity Demand using Kohonem Self-Organizing Maps. *13th International Conference on STATISTICS, MATHEMATICS AND COMPUTATION* , pp. 1-15.
- Pesca e Companhia*. (2019, junho 24). Retrieved from Pesca e Companhia: <https://pescaecia.com.br/2019/06/24/guia-basico-fish-finder/>
- Pickard, G. L., & Emery, W. J. (1990). *Descriptive Physical Oceanography*. New York: Pergamon Press.
- Santos, C. I. (2015). Acústica Submarina Classificação de navios aplicando algoritmos de aprendizagem supervisionada (data mining). *Acústica Submarina*.
- Santos, M. Y., & Ramos, I. (2017). *BUSINESS INTELLIGENCE DA INFORMAÇÃO AO CONHECIMENTO*. Lisboa: FCA.
- Sharma V, A. (2017, março 30). Understanding Activation Functions in Neural Networks. *Understanding Activation Functions in Neural Networks*.
- Tan, L., & Jeang, J. (2013). *Digital Signal Prossecing Fundamentals and Application*. Elsevier.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining, (Firsth Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co.



- Távora, R. G. (2017). *Deteção de Réplicas em Evidências de Áudio usando um Esquema Adaptativo de Audio Fingerprinting*. Brasília: PGEA.TD-123/17.
- Varela, A. J. (2001). Apontamentos de Armas Submarinas e Sistemas de Detecção e Armamento Submarino. *Armas Submarinas e Sistemas de Detecção, Escola Naval*.
- Vasconcelos, J. B. (2015). *PYTHON Algoritmia e Programação*. Lisboa: FCA.
- Versanto, J. (2000, April 18). Neural network tool for data mining: Som toolbox. *Proceeding of symposium on tool environments and development methods for intelligent systems (TOOLMET2000)*, pp. 184-196.
- Witten, I. H., & Frank, E. (2005). *Data Mining Practical Machine Learning Tools and Techniques second edition*. San Francisco: Elsevier.
- Zhang, S. &. (2003, maio). Data Preparation for Data Mining. *Applied Artificial Intelligence*, pp. 17(5-6):375-381.



## Anexos

### Anexo A – Resultados de múltiplos SOM's

Treino de SOM's usando os dados de treino, com *lattice* hexagonal e a inicialização através de PCA.

Treino n.º	Intervalo de escolha de neurónios	Normalização	Treino bruto	Treino fino	Modelo escolhido	TE	QE	Precisão
1	[18,21] X [18,21]	Standard	40	100	2 – 18 X 19	0,19	0,26	18%
2	[18,21] X [18,21]	Min-Max	40	100	16 – 18 X 21	0,20	0,04	29%
3	[18,21] X [18,21]	Nenhuma	40	100	6 – 21 X 19	0,02	2,46	14%
4	[18,21] X [18,21]	Nenhuma	20	50	5 – 21 X 18	0,03	2,32	18%
5	[18,21] X [18,21]	Min-Max	20	50	4 – 21 X 18	0,24	0,04	18%
6	[18,21] X [18,21]	Standard	20	50	20 – 19 X 18	0,14	0,27	25%
7	[9,10] X [9,11]	Standard	20	50	1 – 9 X 10	0,21	0,37	14%
8	[9,10] X [9,10]	Min-Max	20	50	6 – 9 X 10	0,32	0,07	29%
9	[9,10] X [9,10]	Nenhuma	20	50	0 – 10 X 10	0,05	3,66	32%
10	7 X 10	Nenhuma	20	50	0	0,05	4,20	29%
11	7 X 10	Min-Max	20	50	0	0,41	0,07	29%
12	7 X 10	Standard	20	50	0	0,16	0,40	18%

Treino de SOM's usando os dados de treino, com *lattice* retangular e a inicialização através de PCA.

Treino n.º	Intervalo de escolha de neurónios	Normalização	Treino bruto	Treino fino	Modelo escolhido	TE	QE	Precisão
1	[18,21] X [18,21]	Standard	40	100	13 – 18 X 19	0,07	0,13	11%
2	[18,21] X [18,21]	Min-Max	40	100	4 – 19 X 21	0,02	0,02	3%
3	[18,21] X [18,21]	Nenhuma	40	100	9 – 20 X 21	0,03	1,01	18%
4	[18,21] X [18,21]	Nenhuma	20	50	3 – 20 X 18	0,02	1,23	0%
5	[18,21] X [18,21]	Min-Max	20	50	6 – 18 X 18	0,23	0,03	29%
6	[18,21] X [18,21]	Standard	20	50	6 – 21 X 19	0,03	0,13	32%
7	[9,10] X [9,10]	Standard	20	50	0 – 10 X 10	0,14	0,28	18%
8	[9,10] X [9,10]	Min-Max	20	50	1 – 9 X 10	0,12	0,04	14%
9	[9,10] X [9,10]	Nenhuma	20	50	1 – 10 X 10	0,06	2,79	32%
10	8 X 9	Nenhuma	20	50	0	0,05	4,20	18%
11	8 X 9	Min-Max	20	50	0	0,22	0,05	18%
12	8 X 9	Standard	20	50	0	0,22	0,32	18%

## Anexo B – Tabela de ‘presença’ do SOM selecionado

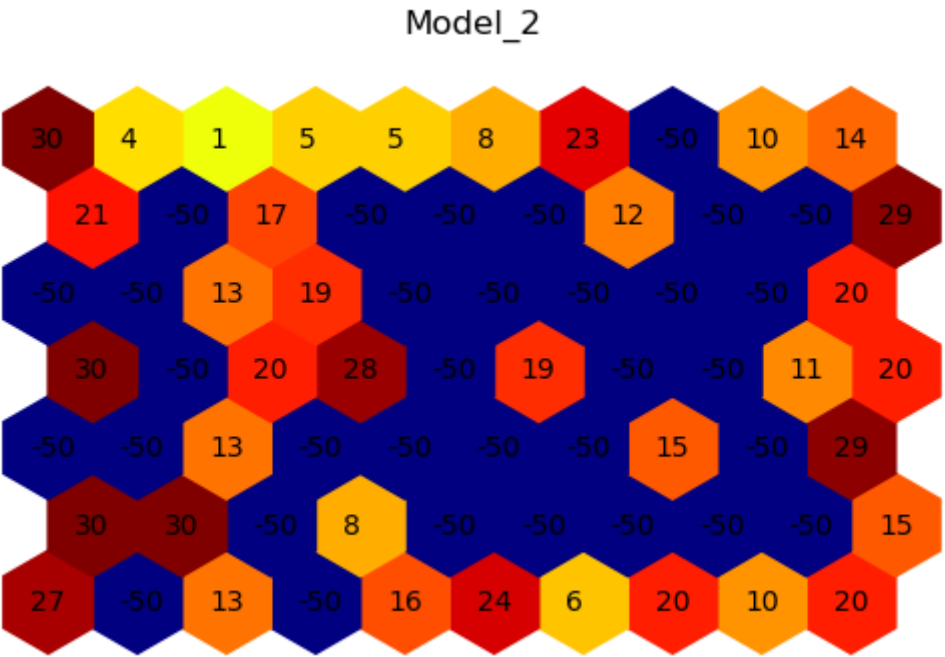
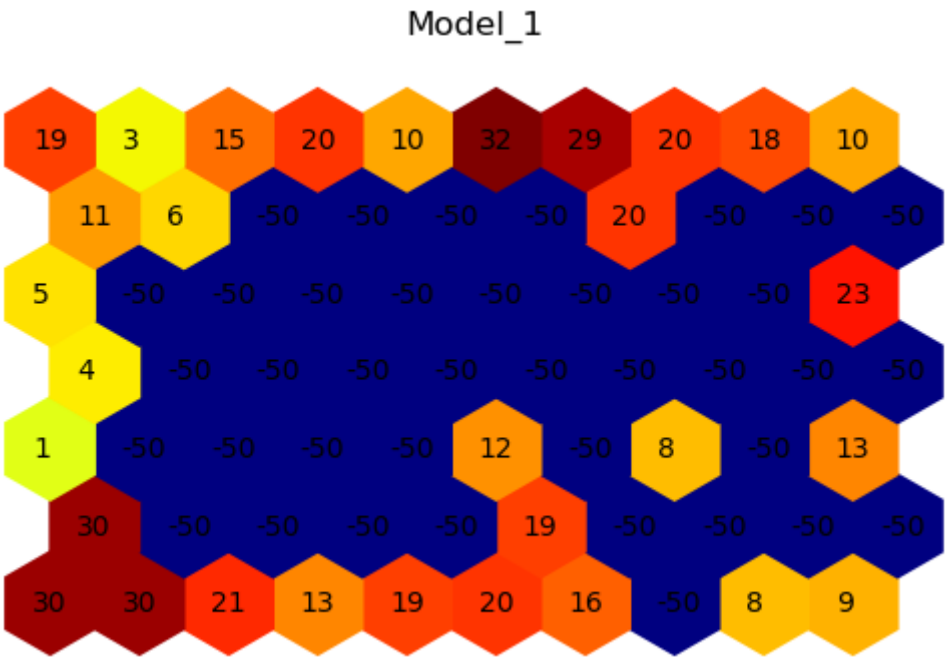
Na seguinte tabela, pode-se ver com sombreado amarelo a seleção efetuada por neurónio pelo algoritmo criado, a sombreado a azul a opção que poderia ser tomada como alternativa e também as restantes classes alocadas a cada um dos neurónios, juntamente com a confiança de cada classe por nó.

Nó do SOM	Ficheiro n.º - contagem de treino/ confiança	Ficheiro - conta- gem de treino/ confiança	Ficheiro - conta- gem de treino/ confiança	Ficheiro - con- tagem de treino/ confi- ança
0	10 - 4/0.5	20 - 4/0.14	32 - 2/0.17	
1	32 - 2/0.17			
2	20 - 4/0.14	29 - 3/0.75		
3	Nenhum			
4	18 - 4/1			
5	10 - 4/0.5	14 - 4/1	33 - 4/1	
6	Nenhum			
7	12 - 3/0.75			
8	8 - 1/0.25	12 - 1/0.25	23 - 4/1	26 - 4/1
9	3 - 3/0.75	11 - 4/1	19 - 4/0.2	20 - 4/0.14
10	15 - 2 /0.5	29 - 1/0.25		
11	Nenhum			
12	3 - 1/0.25			
13	Nenhum			
14	Nenhum			
15	Nenhum			
16	Nenhum			
17	Nenhum			
18	20 - 1/0.03			
19	5 - 2/0.5			
20	Nenhum			
21	Nenhum			
22	Nenhum			
23	Nenhum			
24	Nenhum			
25	Nenhum			
26	Nenhum			
27	Nenhum			
28	13 - 1/0.083	19 - 3/0.15		
29	1 - 4/1	5 - 2/0.5	31 - 4/1	
30	15 - 1/0.25	20 - 8/0.29		
31	Nenhum			
32	Nenhum			

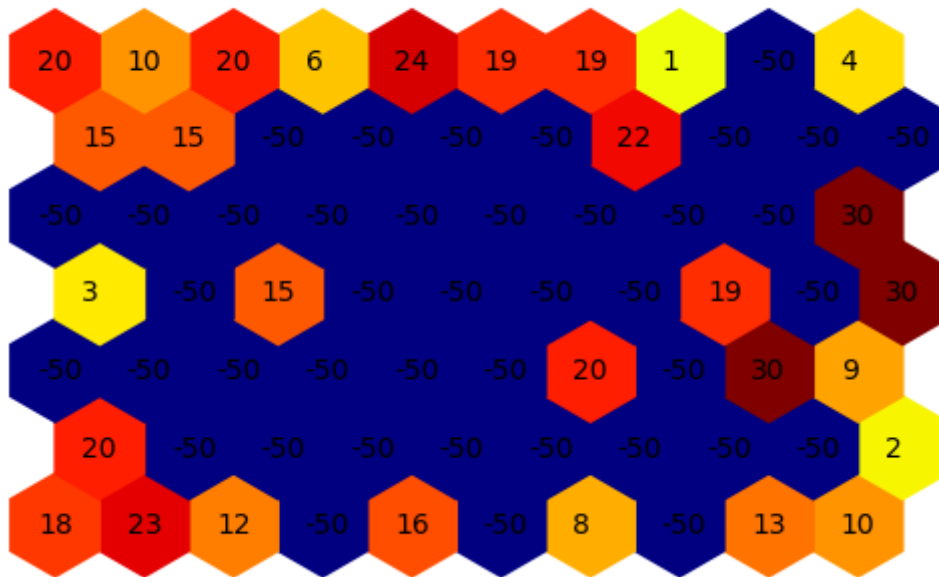
33	Nenhum			
34	Nenhum			
35	27 – 4/1	32 – 4/0.33		
36	Nenhum			
37	Nenhum			
38	13 – 1/0.08	20 – 3/0.11		
39	19 – 7/0.35	22 – 4/1		
40	20 – 4/0.14			
41	Nenhum			
42	Nenhum			
43	Nenhum			
44	Nenhum			
45	2 – 4/1	9 – 4/0.5	13 – 8/0.66	
46	Nenhum			
47	Nenhum			
48	19 – 1/0.05			
49	4 – 4/1	7 – 4/1	17 – 4/1	
50	6 – 4/1	15 – 1/0.25		
51	Nenhum			
52	Nenhum			
53	Nenhum			
54	8 – 3/0.75			
55	Nenhum			
56	Nenhum			
57	Nenhum			
58	Nenhum			
59	21 – 1/0.25			
60	24 – 4/1	25 – 4/1		
61	19 – 3/0.15			
62	19 – 1/0.05			
63	16 – 4/1	19 – 1/0.05	28 – 4/1	
64	13 – 2/0.16			
65	9 – 4/0.5			
66	30 – 12/0.3			
67	30 – 24/0.6			
68	Nenhum			
69	21 – 3/0.75	30 – 4/0.1	32 – 4/0.3	

Anexo C – Mapas de classificação dos SOM's

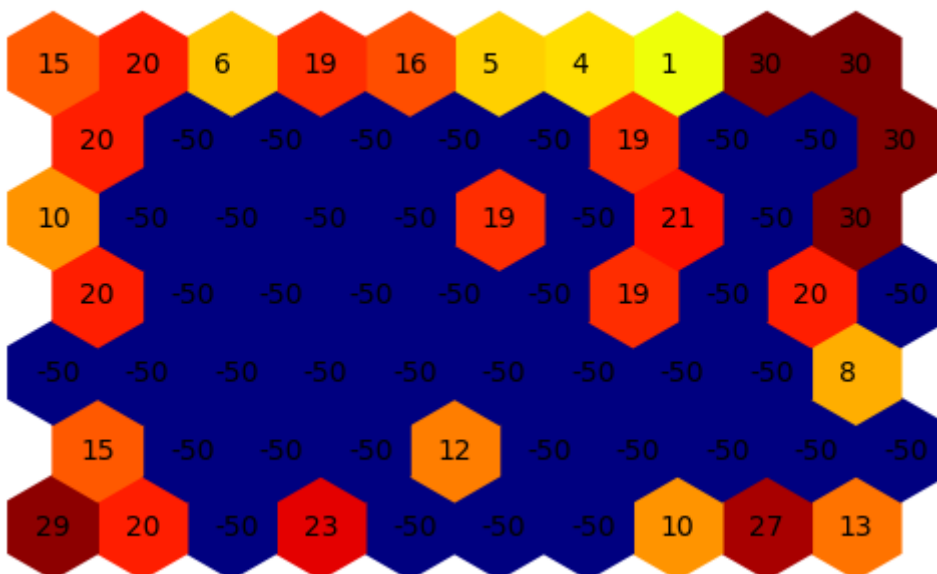
Mapas de classificação de 10 mapas treinados, com dimensão  $7 \times 10$ , grelha hexagonal, 20 iterações de treino bruto e 50 iterações de treino fino e inicialização aleatória.



Model\_3

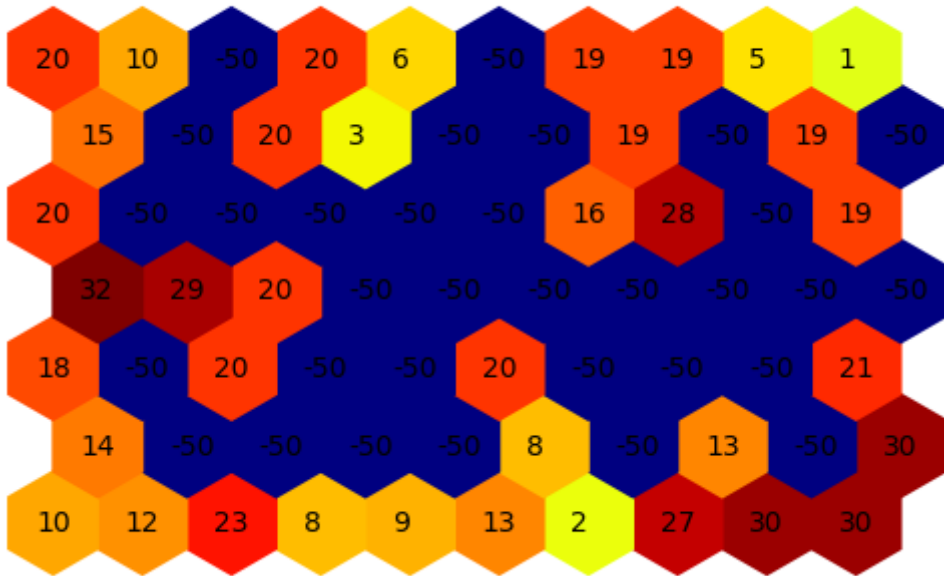


Model\_4

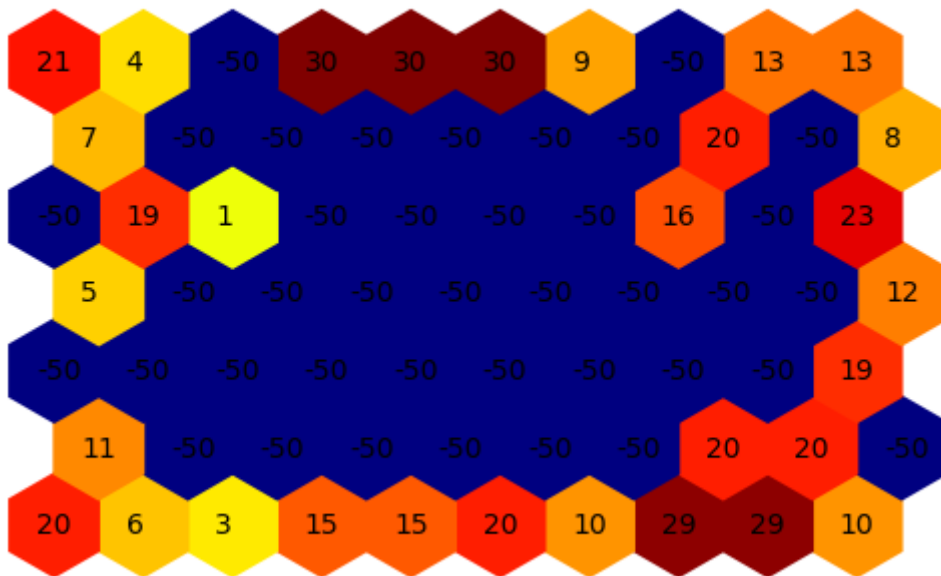




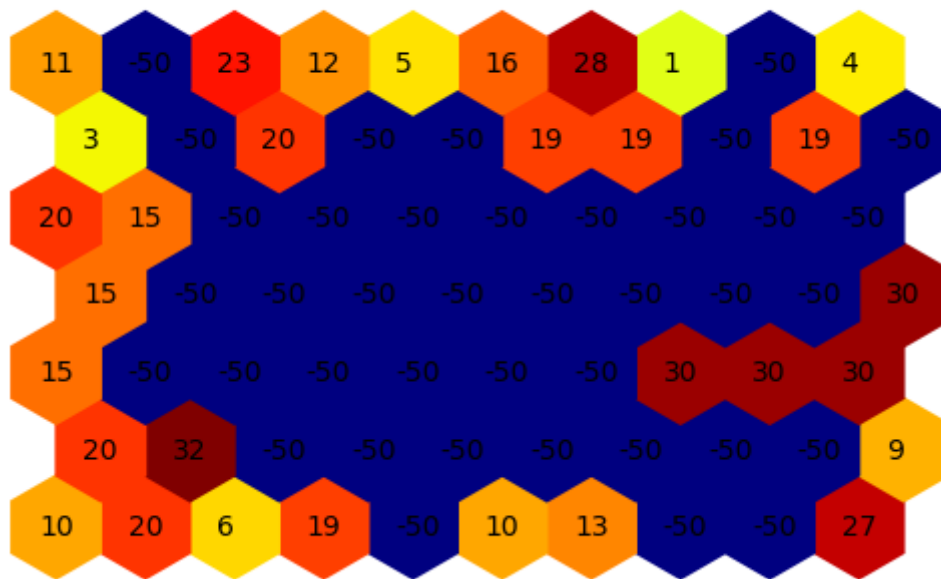
Model\_5



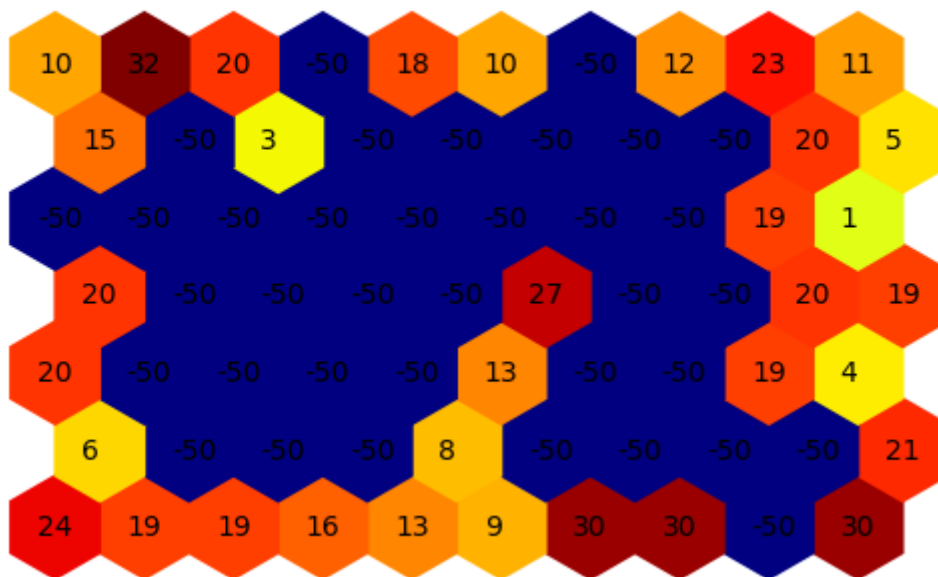
Model\_6



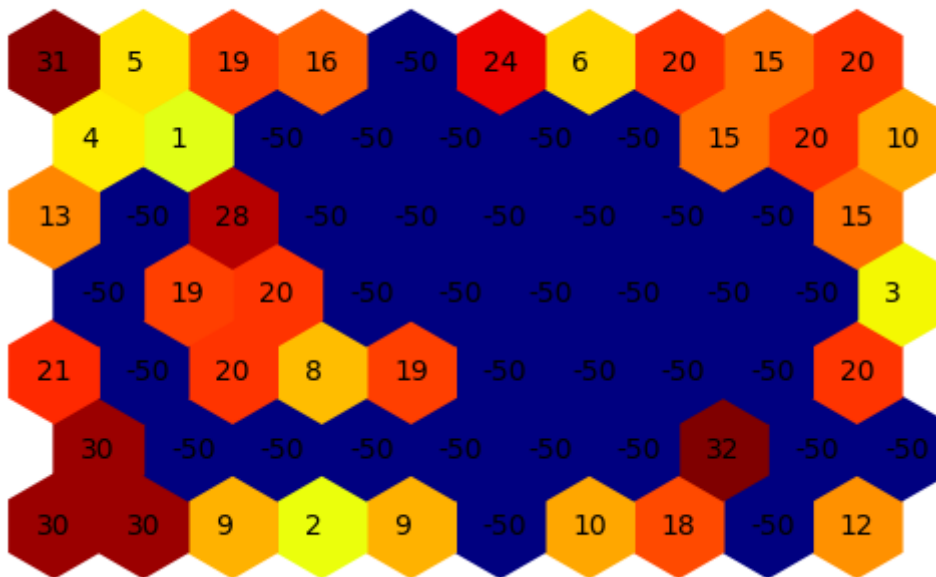
Model\_7



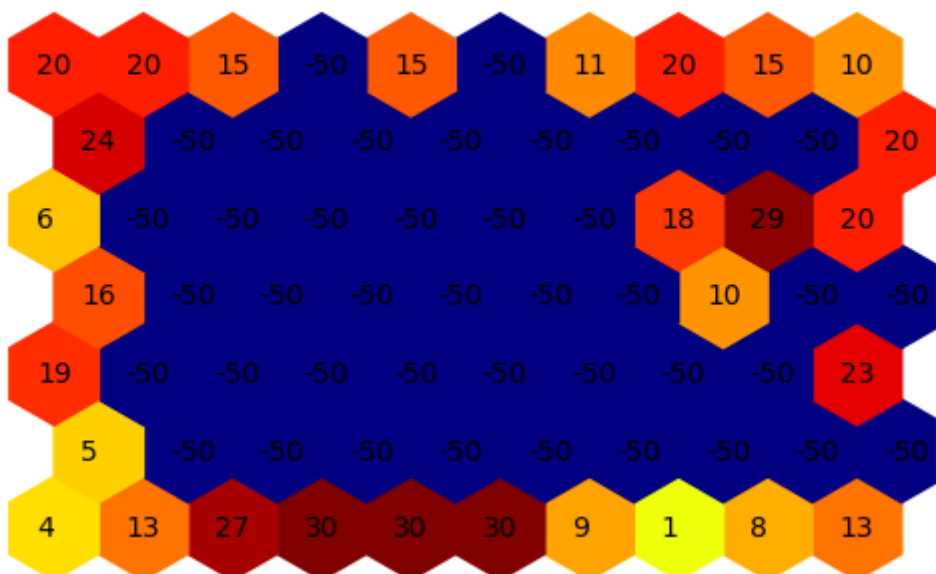
Model\_8



Model\_9

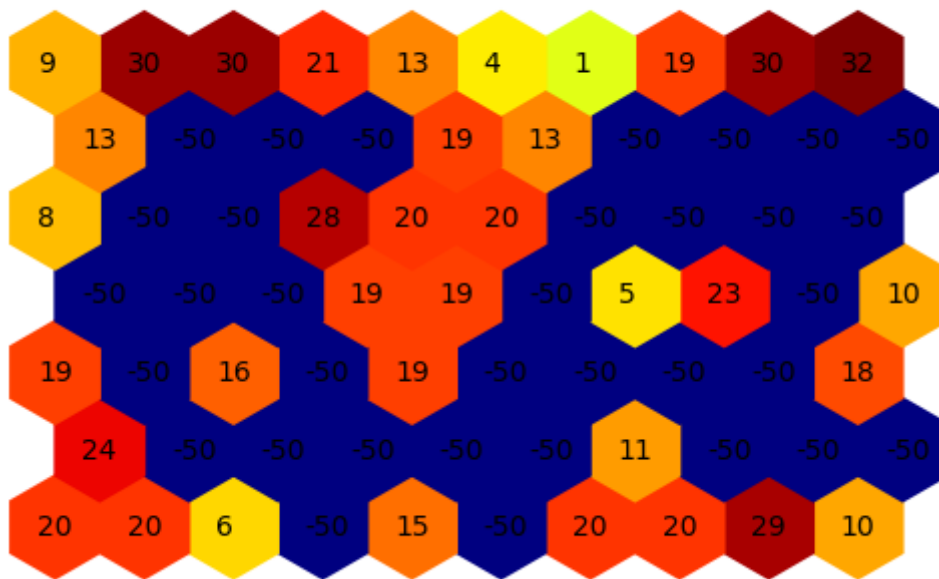


Model\_10



Mapa de componente de um mapa SOM treinado com dimensão  $7 \times 10$ , grelha hexagonal, 20 iterações de treino bruto e 50 iterações de treino fino e inicialização por PCA juntamente com o resultado da precisão para os dados de teste.

Model\_11



Modelo n.º	Precisão de treino	Precisão de teste	Erro topográfico	Erro de quantização
11	63,3%	51,7%	0,15	0,06

## Anexo D – Resultado de múltiplas RNA's

Resultados para o treino de RNA's com uma camada de entrada de dimensão 193 (n.º de variáveis dependentes), 1 ou 2 camadas escondidas e uma camada de saída com 34 neurónios (33 navios e uma mais para a classificação desconhecida) validando com os dados de validação.

N.º de neurónios das Camadas escondidas	Ativação dos neurónios internos	Ativação dos neurónios de saída	N.º de iterações	Precisão
Normalização: Min-Max				
4	ReLU	softmax	100	32%
8	ReLU	softmax	100	18%
16	ReLU	softmax	100	25%
32	ReLU	softmax	100	32%
64	ReLU	softmax	100	43%
128	ReLU	softmax	100	29%
Normalização: Standard				
4	ReLU	softmax	50	25%
8	ReLU	softmax	50	32%
16	ReLU	softmax	50	14%
32	ReLU	softmax	50	18%
64	ReLU	softmax	50	25%
128	ReLU	softmax	50	18%
Normalização: Nenhuma				
4	ReLU	softmax	200	25%
8	ReLU	softmax	200	32%
16	ReLU	softmax	200	21%
32	ReLU	softmax	200	18%
64	ReLU	softmax	200	39%
128	ReLU	softmax	200	21%

Resultados obtidos para o treino de RNA's com 64 neurónios na camada interna, sem se efetuar a normalização dos dados.

- n.º de iterações = 200

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	70.7	35	4.73	6.69
2	89.7	38.3	1.68	5.92
3	86.2	40	2.23	6.86
4	96.5	43.3	0.57	5.93
5	79.3	45	3.34	7.86
6	96.6	41.7	0.57	6.13
7	98.3	40	0.30	5.13
8	94.8	36.7	0.84	6.24
9	96.6	33.3	0.57	5.92
10	87.5	35	2.03	6.19

- n.º de iterações = 100

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	93.1	43.3	1.18	6.39
2	91.4	36.7	1.42	5.37
3	96.6	35	0.60	5.69
4	91.4	41.7	1.43	5.32
5	93.1	33.3	1.15	5.61
6	72	45	4.51	5.69
7	90.9	41.7	1.44	5.06
8	96.1	38.3	0.61	4.94
9	94.8	43.3	0.88	3.87
10	90.9	46.7	1.44	3.95

- n.º de iterações = 50

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	96.1	38.3	0.51	3.59
2	94.4	40	0.74	4.98
3	91.3	40	1.17	5.70
4	88.8	36.7	1.61	5.28
5	89.2	38.3	1.66	4.76
6	84.9	43.3	2.02	5.01
7	91.8	40	1.10	4.06
8	88.4	40	1.61	4.83
9	84.1	40	2.02	4.64
10	91.8	48.3	1.13	3.48

Resultados obtidos para o treino de RNA's com 64 neurónios na camada interna, com os dados normalizados por Min-Max.

- n.º de iterações = 200

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	87.1	41.7	0.72	2.65
2	90.5	41.7	0.64	2.72
3	88.8	40	0.68	2.79
4	88.3	41.7	0.64	2.76
5	86.2	38.3	0.76	2.65
6	90.1	40	0.66	2.77
7	87.9	41.7	0.70	2.78
8	83.2	38.3	0.75	2.70
9	89.2	41.7	0.71	2.83
10	89.2	41.7	0.65	2.95

- n.º de iterações = 250

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	97.4	43.3	0.37	3.12
2	95.3	45	0.42	3.05
3	96.1	46.7	0.42	2.89
4	95.7	40	0.42	3.02
5	95.7	45	0.38	3.04
6	94.8	46.7	0.43	3.01
7	96.1	46.7	0.36	3.22
8	93.1	41.7	0.46	3.01
9	94.4	46.7	0.42	3.03
10	94	41.7	0.42	2.96

- n.º de iterações = 150

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	0.72	41.7	1.06	2.58
2	0.72	30.0	1.10	2.56
3	0.70	26.7	1.13	2.61
4	0.71	33.3	1.11	2.72
5	0.68	35	1.18	2.68
6	0.74	28.3	1.08	2.58
7	0.75	33.3	1.00	2.60
8	0.73	36.7	1.07	2.54
9	0.75	33.3	1.10	2.59
10	0.73	28.3	1.02	2.77

Resultados obtidos para o treino de RNA's com 64 neurónios na camada interna, com os dados normalizados Standard.

- n.º de iterações = 200

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	100	41.7	0.04	3.67
2	100	41.7	0.04	4.14
3	100	40	0.04	4.19
4	100	40	0.04	3.81
5	100	41.7	0.04	3.71
6	100	41.7	0.04	3.63
7	100	40	0.04	3.65
8	100	40	0.04	4.05
9	100	40	0.04	3.78
10	100	40	0.04	3.99

- n.º de iterações = 100

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	98.3	45	0.26	2.56
2	98.3	45	0.28	3.01
3	98.7	45	0.26	2.96
4	98.3	45	0.27	3.08
5	99.1	43.3	0.26	3.27
6	98.3	45	0.27	2.77
7	98.3	45	0.25	3.03
8	98.3	45	0.25	2.91
9	98.3	45	0.28	3.04
10	98.7	46.7	0.28	2.83

- n.º de iterações = 50

Modelo n.º	Precisão de treino	Precisão de teste	Erro de treino	Erro de teste
1	71.1	35	1.16	2.86
2	69	33.3	1.17	2.82
3	66.8	28.3	1.19	2.69
4	68.5	35	1.20	2.72
5	67.7	28.3	1.21	2.57
6	70.7	33.3	1.18	2.81
7	69.8	26.7	1.20	2.65
8	69.4	31.7	1.20	2.68
9	71.2	26.7	1.17	2.62
10	70.7	23.3	1.21	2.77



## Anexo E – Programas

### Carregar as gravações e extrair

```
import numpy as np
np.random.seed(1001)
import librosa
import glob

all_wav_files = glob.glob('C:\\Users\\Cláudio António\\PycharmProjects\\teste2\\*.wav')
#var. onde está gravado todo
all_wav_files_valid = glob.glob('C:\\Users\\Cláudio António\\PycharmProjects\\teste2\\valid\\*.wav')

def extract_feature(file): # função que extrai os recursos dos sons
    X, sample_rate = librosa.load(file)
    stft = np.abs(librosa.stft(X))
    mfccs = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
    chroma = np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T, axis=0)
    mel = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T, axis=0)
    contrast = np.mean(librosa.feature.spectral_contrast(S=stft, sr=sample_rate).T,
                      axis=0)
    tonnetz = np.mean(librosa.feature.tonnetz(y=librosa.effects.harmonic(X),
                                             sr=(sample_rate.T, axis=0)
    return mfccs, chroma, mel, contrast, tonnetz

def parse_audio_files(file): #função que retorna um vector com os recursos dos áudios
    mfccs, chroma, mel, contrast, tonnetz = extract_feature(file) #Chama a função que
    extrai os recursos dos audios
    ext_features = np.hstack([mfccs, chroma, mel, contrast, tonnetz]) #vector com os
    recursos
    print("extract %s features done" % (file))
    return np.array(ext_features)

data = []
#parent_dir='PycharmProjects'
for file in all_wav_files: #ciclo para percorrer todos os ficheiros de som
    audio = parse_audio_files (file)
    data.append(audio) #adiciona o vector dos recursos na variável data

audio_class= [] # será a variável(vector) que receberá as classificações dos dados
for file in all_wav_files:
    audio_class.append(file.split('.')[0].split("\\")[-1]) #separação da string

data = np.c_[data, audio_class] # concatenar os vectores de recursos com o vector das
classif. na var. data
```

```
import pickle as pkl
with open('data.pkl', 'wb') as f:    #criar um ficheiro com os dados já organizados
    pkl.dump(data, f)
```

---

### **Criação de SOM's**

```
import pickle as pkl
import pandas as pd
# Começar por carregar os dados

with open('data.pkl','rb') as f:
    data = pd.DataFrame(pkl.load(f))

X_data = data.drop(columns=193).astype(float)
Y_data = data[193]

d = dict(zip(X_data.columns, ['feature{ }'.format(x) for x in range(len(X_data.T))]))
X_data = X_data.rename(columns=d)
labels = X_data.columns

import os
import glob
from sklearn.externals import joblib # útil para guardar modelos gerados aleatoriamente

del_old_files = glob.glob("./sommodels\\som_model*")
for f in del_old_files:
    os.remove(f)

# Este Código serve se pretendemos calcular automaticamente a dimensão do SOM
from sompy.sompy import SOMFactory

som = SOMFactory().build(X_data.values, mapsize=[5,8], initialization='random',
                        component_names=labels, lattice="hexa")

print(som.calculate_map_size(lattice= 'hex'))

import time # Para calcular o tempo necessário
import random
stime = time.time()
for i in range(20): # Criar vários modelos com dimensões de grelha aleatórios
    som = SOMFactory().build(data.values,
                            mapsize=[random.choice(list(range(7,8))),random.choice(list(range(10,11)))],
                            initialization='pca', component_names=labels,
                            lattice="hexa",normalization='var')
    som.train(n_job=1, verbose=False, train_rough_len=40, train_finetune_len=100)
```

```

joblib.dump(som,"./Segment_analysis_with_SOM\\hexa_som_models\\
    hexa_som_model_{ }.joblib".format(i)) # salvar o modelo
print(i)
print("took %.2f"%(time.time()-stime)+" sec")

models = glob.glob("./Segment_analysis_with_SOM\\
    hexa_som_models\\hexa_som_model*") # Criar uma lista de todos os modelos

te_all = []
qe_all = []
for model in models:
    som = joblib.load(model)
    te_all.append(som.calculate_topographic_error())
    qe_all.append(som.calculate_quantization_error())

print(te_all, qe_all)

import matplotlib.pyplot as plt # to plot

'''Gráfico de dispersão para os erros topográficos e de quantização de todos os modelos'''
plt.scatter(te_all,qe_all)
for i in range(len(te_all)):
    plt.text(te_all[i],qe_all[i],str(i))
plt.show()

''' Se todos os erros topográficos forem baixos basta escolher o modelo com menor erro
de quantização'''
selec_model = qe_all.index(min(qe_all))
#alternativa (distância à origem)
#norm_qe_te = np.linalg.norm(np.array([np.asarray(te_all),np.asarray(qe_all)]),axis=0)
#selec_model = min(np.where(norm_qe_te==min(norm_qe_te))[0])

som_chosen = joblib.load("./Segment_analysis_with_SOM\\
    hexa_som_models\\hexa_som_model_{ }.joblib".format(selec_model))
print("chosen model number_"+str(selec_model))

'''Salvar modelo na raíz'''
joblib.dump(som_chosen,"./Segment_analysis_with_SOM\\
    hexa_chosen_som_model.joblib")

```

---

### **Algoritmo para apoio à decisão e visualização de mapas de Hits**

```
import pandas as pd
from sklearn.externals import joblib ## útil para guardar modelos gerados aleatoriamente

import pickle as pkl
import numpy as np
import pandas as pd

''' Carregar as components dos audio'''
with open('.\X_data.pkl','rb') as f:
    data = pd.DataFrame(pkl.load(f))

'''recuperar modelo escolhido'''
som = joblib.load('./hexa_chosen12.joblib.')

'''Visualização do HitsMap'''

import matplotlib.pyplot as plt # to plot
from sompy.visualization.bmuhits import BmuHitsView

vhts = BmuHitsView(3,3,"Hits Map",text_size=7)
vhts.show(som, anotate=True, onlyzeros=False, labelsiz=7, cmap="autumn",
          logarithmic=False)
plt.show()

node_matrix_location = pd.DataFrame(getattr(som.codebook,'matrix')) # localização de
cada nó

Node_assignment_of_data=pd.DataFrame(som.project_data(data), index=data.index)

Descriptive_data = data.copy()
Descriptive_data['Node']=Node_assignment_of_data

''' Carregar o y_train '''
with open('.\Y_data.pkl','rb') as f:
    target = pd.DataFrame(pkl.load(f))

Descriptive_data['File']=target.File_name.copy()

Files_in_data = np.unique(Descriptive_data.FileNum)
Nodes_in_data = np.unique(Descriptive_data.Node)

save_classification = pd.DataFrame(Nodes_in_data,columns=pd.Index(['Node']))
```

```

save_classification['classification'] = -1

for node in Nodes_in_data:
    Files_in_selected_Node = Descriptive_data.FileNum[Descriptive_data.Node ==
                                                         node]

    Files_unique = np.unique(Files_in_selected_Node)
    confidence_save = []
    for file in Files_unique:
        confidence = np.sum(Files_in_selected_Node == file) /
                        np.sum(Descriptive_data.FileNum==file)
        confidence_save = np.append(confidence_save,confidence)
        print(node,file,confidence)
    save_classification.classification[save_classification.Node == node] =
        str(Files_unique[confidence_save == np.max(confidence_save)])
    print('File_classification = '+ str(Files_unique[confidence_save ==
        np.max(confidence_save)]))

node_prediction = som.project_data(np.reshape(data.loc[5].values.T,(1,-1)))
print(node_prediction)
location = (Descriptive_data['Node'] == float(node_prediction))
Descriptive_data['FileNum'].loc[location]

def classification(data_used):
    classification = []

    Node = som.project_data(data_used)
    for node in Node:
        nodes_in_data = (Descriptive_data.Node == node)
        Data_select = Descriptive_data.FileNum[nodes_in_data]
        Files = np.unique(Data_select)
        File_probality = []
        counts = []
        confidences = []
        for file in Files:
            count = np.sum(Data_select == file)
            length_FileNum = np.sum(Descriptive_data.FileNum == file)
            confidence = count/length_FileNum
            File_probality = np.append(File_probality,
                'File = '+ str(file) +' / '+
                'counts = ' + str(count) +' / '+
                'confidence = ' + str(confidence) + ':::')
            counts = np.append(counts,count)
            confidences = np.append(confidences,confidence)

        classification = np.append(classification,str(File_probality))
        if np.sum(nodes_in_data) == 0:
            classification = 'None'

```

```

        counts = 'None'
        confidences = 'None'
        selected_File = 'None'
    else:
        max_counts = np.max(counts)
        selected_File = Files[counts==max_counts]
    return classification, counts, confidences, selected_File[0]

single_choice = np.reshape(data.iloc[1].values.T,(1,-1))

print(classification(single_choice))

decision = []
for i in range(len(data)):
    single_choice = np.reshape(data.iloc[i].values.T, (1, -1))
    predicted = classification(single_choice)[3]
    real = target.File_target.iloc[i]
    if real == predicted:
        text = 'correct'
    else:
        text = 'wrong'
    decision = np.append(decision, text)
    print(str(predicted)+'->'+str(real)+' : '+ text)

Error_in_train = np.sum(decision == 'correct')/len(data)

print(str(Error_in_train*100)+'%')

with open('./Segment_analysis_with_SOM\\X_test.pkl','rb') as f:
    test = pd.DataFrame(pk1.load(f))

with open('./Segment_analysis_with_SOM\\y_test.pkl','rb') as f:
    target_test = pd.DataFrame(pk1.load(f))

decision = []
for i in range(len(test)):
    single_choice = np.reshape(test.iloc[i].values.T, (1, -1))
    predicted = classification(single_choice)[3]
    real = target_test.File_target.iloc[i]
    if real == predicted:
        text = 'correct'
    else:
        text = 'wrong'
    decision = np.append(decision, text)
    print(str(predicted)+'->'+str(real)+' : '+ text)

```

```
Error_in_test = np.sum(decision == 'correct')/len(test)
print(str(Error_in_test*100)+'%')
```

---

### **Treino das Redes Neurais Artificiais**

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

X_train = pd.read_pickle('./Segment_analysis_with_SOM\\X_train.pkl')
y_target = pd.read_pickle('./Segment_analysis_with_SOM\\y_train.pkl')

y_class = np.asarray(y_target.File_target)

y_train = []

for value in y_class:
    y_train = np.append(y_train,np.int8(float(value)))

plt.close('all')

X_valid = pd.read_pickle('./Segment_analysis_with_SOM\\X_valid.pkl')
y_target_valid = pd.read_pickle('./Segment_analysis_with_SOM\\y_valid.pkl')

y_class_valid = np.asarray(y_target_valid.File_target)

y_valid = []

for value in y_class_valid:
    y_valid = np.append(y_valid,np.int8(float(value)))

from keras.models import Sequential
from keras.layers import Dense

from keras.layers.advanced_activations import ReLU

model = Sequential()

model.add(Dense(64,input_dim=193, kernel_initializer='normal'))
model.add(ReLU())
"""camada de saída terá 34 neurónios (33 navios distintos) e será baseado em probabilidade
por classe"""
model.add(Dense(34,activation='softmax', kernel_initializer='normal'))
# parâmetros para treinar o modelo
```

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

```
history = model.fit(X_train.values, y_train, validation_data=(X_valid.values,y_valid),  
epochs=200, batch_size=len(X_train.values), verbose=0)
```

```
'''será que está a fazer overfitting?'''
```

```
X_test = pd.read_pickle('./Segment_analysis_with_SOM\\X_test.pkl')  
y_target_test = pd.read_pickle('./Segment_analysis_with_SOM\\y_test.pkl')
```

```
y_class_test = y_target_test.File_target
```

```
y_test = []
```

```
for value in y_class_test:  
    y_test = np.append(y_test, np.int8(float(value)))
```

```
val_test_loss,val_test_acc = model.evaluate(X_valid.values, y_valid)  
val_train_loss,val_train_acc = model.evaluate(X_train.values, y_train)  
print(val_test_loss, val_test_acc)  
print(val_train_loss, val_train_acc)
```

```
plt.figure(2)  
plt.subplot(2,1,1)  
plt.plot(history.history['acc'], label='Train_accuracy')  
plt.plot(history.history['val_acc'], label='Valid_accuracy')  
plt.legend()
```

```
plt.subplot(2,1,2)  
plt.plot(history.history['loss'], label='Train_loss')  
plt.plot(history.history['val_loss'], label='Valid_loss')  
plt.legend()  
plt.show()
```

```
y_predict=model.predict(X_valid)  
np.argmax(y_predict,axis=1)  
y_valid  
y_train_predict = model.predict(X_train)  
y_train_predict  
np.argmax(y_train_predict,axis=1)
```

```
#model.save("./Manual_Division\\models_per_group\\ann_model_{ }.h5".format(Group))
```